



PRUEBA DE CONCEPTO DE NUEVAS APLICACIONES  
CON OCULUS RIFT Y UNITY:  
UN PASEO VIRTUAL POR LA UC<sub>3</sub>M

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA  
ESCUELA POLITÉCNICA SUPERIOR

AUTOR: JAVIER DE LOS REYES GUÍO  
TUTORES: YAGO SÁEZ ACHAERANDIO  
ALEJANDRO BALDOMINOS GÓMEZ



## AGRADECIMIENTOS

En primer lugar, quiero mostrar mi agradecimiento a mis padres y a mi hermano, que han estado apoyándome desde que empecé mi andadura en esta universidad, aconsejándome y ayudándome en los momentos difíciles que han surgido durante mi carrera.

En segundo lugar, a mis compañeros y amigos de la universidad, por todos esos momentos de alegrías y penas que he vivido junto a ellos. Sin ellos nada de esto habría sido posible.

A aquellos compañeros de clase con los que he tenido el placer de colaborar durante las prácticas y las asignaturas de la carrera, gracias.

Al resto de mis amigos, los que me soportan cada fin de semana, gracias por compartir conmigo los momentos de éxito y por estar ahí apoyándome cuando lo he pasado mal.

Y por último, pero no menos importante, agradecer enormemente su labor a mis tutores, Yago y Alejandro, por 'darme caña' durante todo este tiempo, por responder a mis dudas y por guiarme a lo largo de este trabajo.

Muchas gracias a todos.



## RESUMEN

En el presente trabajo se desarrolla un prototipo de visita virtual al edificio Sabatini de la Escuela Politécnica Superior, implementado con el motor Unity 3D y modelado en Blender, que utiliza, a su vez, el hardware de realidad virtual Oculus Rift para ofrecer la mayor inmersión posible del usuario en el entorno. Este documento refleja las decisiones de diseño, funcionalidades y pruebas realizadas con la aplicación y el módulo de realidad virtual, aportando una serie de valoraciones acerca de los satisfactorios resultados obtenidos en la encuesta realizada y sobre el desarrollo del trabajo en general, y finalizando con unas conclusiones sobre el proceso completo de este trabajo y unos comentarios acerca de las mejoras que pueden introducirse en este paseo virtual por la UC3M.



## ABSTRACT

In this bachelor thesis, a virtual visit prototype to the Sabatini building at EPS is developed, implemented with Unity 3D engine and shaped in Blender, which uses the virtual reality hardware Oculus Rift to offer as much immersion as possible. This document reflects design decisions, functionalities and tests realized with the application, providing a series of valuations about the satisfactory results obtained in the survey and on the development of the whole thesis in general, and ending with some conclusions about the whole process of this bachelor thesis and a few comments on what improvements can be made in this virtual tour of the UC3M.



## TABLA DE CONTENIDOS

Agradecimientos.....	2
Resumen.....	3
Abstract.....	4
Índice de ilustraciones.....	7
Índice de tablas.....	9
Introducción.....	13
Objetivo y motivación.....	13
Estructura del documento.....	14
Glosario de términos empleados.....	15
Estado del arte.....	17
Recorrido histórico de la realidad virtual <sup>[8]</sup> .....	17
Estudio de alternativas.....	21
Hardware de realidad virtual.....	22
Motores gráficos y SDK.....	29
Análisis de la propuesta.....	35
Análisis de viabilidad del sistema.....	35
Identificación de stakeholders.....	35
Estudio de la situación actual.....	36
Definición de casos de uso.....	37
Requisitos del sistema.....	41
Requisitos funcionales.....	42
Requisitos no funcionales.....	47
Diseño e implementación de la propuesta.....	49
Diseño lógico.....	49
Diagramas de clases.....	49
Flujo de la aplicación mediante estados.....	50
Matriz de trazabilidad de la relación casos de uso-requisitos.....	54
Diseño físico.....	54
Fases de implementación.....	56



Fase de modelado del edificio Sabatini.....	56
Fase de desarrollo de la aplicación en Unity 3D .....	62
Validación y evaluación.....	66
Validación de requisitos .....	66
Evaluación.....	72
Gestión del proyecto.....	76
Planificación inicial.....	76
Planificación final .....	77
Presupuesto .....	78
Conclusiones.....	80
Comentarios finales .....	80
Líneas futuras.....	81
Referencias .....	82
Anexo I: Guía rápida de configuración del sistema .....	86
Anexo II: Diagramas de clases .....	88
Anexo III: Diagramas de planificación.....	94

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Estereoscopio [17] .....	17
Ilustración 2: Gafas 3D basadas en el concepto de 'anaglifo' [18] .....	17
Ilustración 3: Sistema de 'anaglifos' [19] .....	18
Ilustración 4: Cartel promocional de Sensorama [20] .....	18
Ilustración 5: Guante de VR de Gary Grimes [21] .....	19
Ilustración 6: 'VIVED' [22] .....	20
Ilustración 7: Casco VR NASA-Ames [23] .....	20
Ilustración 8: De izquierda a derecha: Virtual Boy, SEGA VR, Wii, Kinect, Oculus Rift [24][25][26][27][28] .....	20
Ilustración 9: Captura de Bioshock Infinite [29] .....	21
Ilustración 10: Project Morpheus [30] .....	22
Ilustración 11: Avegant Glyph [31] .....	23
Ilustración 12: Avegant Glyph Virtual Retina Display [32] .....	24
Ilustración 13: InfinitEye [33] .....	25
Ilustración 14: Oculus Rift DK1 [34] .....	26
Ilustración 15: Oculus Rift DK2 [35] .....	26
Ilustración 16: Comparación del 'screen door effect' en Rift DK1 y DK2 [36] .....	27
Ilustración 17: Interfaz Unity 3D [37] .....	30
Ilustración 18: Interfaz de Unreal Engine 4 [38] .....	31
Ilustración 19: Interfaz de CryEngine [39] .....	32
Ilustración 20: Prototipo de piso .....	58
Ilustración 21: Prototipo de aula informática .....	59
Ilustración 22: Aula informática .....	59
Ilustración 23: Tarima y pizarras del aula de clase .....	60
Ilustración 24: Techo curvo del aula de clases .....	61
Ilustración 25: Asientos del aula de clases .....	61
Ilustración 26: Resultados de la primera pregunta de la encuesta .....	72
Ilustración 27: Resultados de la segunda pregunta de la encuesta .....	73
Ilustración 28: Resultados de la tercera pregunta de la encuesta .....	73
Ilustración 29: Resultados de la cuarta pregunta de la encuesta .....	74



Ilustración 30: Resultados de la quinta pregunta de la encuesta .....	74
Ilustración 31: Diagrama de clases de relación jugador-niveles.....	88
Ilustración 32: Diagrama de clases de relación Outerscene-POI .....	89
Ilustración 33: Diagrama de clases para piso1.....	90
Ilustración 34: Diagrama de clases de la relación piso2-aulainformatica.....	91
Ilustración 35: Diagrama de clases de la relación piso3-aula clase .....	92
Ilustración 36: Diagrama de planificación inicial estimada.....	94
Ilustración 37: Diagrama de planificación final.....	94





## ÍNDICE DE TABLAS

Tabla 1: Especificaciones del display de VR .....	28
Tabla 2: Plantilla de casos de uso .....	37
Tabla 3: CU-01 - Inicio de la aplicación.....	38
Tabla 4: CU-02 – Cierre de la aplicación.....	38
Tabla 5: CU-03 – Primera carga del nivel principal .....	38
Tabla 6: CU-04 – Movimiento del jugador por el nivel actual.....	39
Tabla 7: CU-05 – Movimiento de la cámara de realidad virtual .....	39
Tabla 8: CU-06 – Activación de un ítem de cambio de nivel.....	39
Tabla 9: CU-07 – Activación de ruta de <i>pathfinding</i> .....	40
Tabla 10: CU-08 – Reproducción y controles de escenarios POI .....	40
Tabla 11: CU-09 – Carga de nivel (excepto POI).....	40
Tabla 12: Plantilla de requisitos.....	41
Tabla 13: RF-001.....	42
Tabla 14: RF-002 .....	42
Tabla 15: RF-003.....	43
Tabla 16: RF-004 .....	43
Tabla 17: RF-005.....	43
Tabla 18: RF-006 .....	43
Tabla 19: RF-007 .....	44
Tabla 20: RF-008 .....	44
Tabla 21: RF-009 .....	44
Tabla 22: RF-010 .....	44
Tabla 23: RF-011.....	45
Tabla 24: RF-012 .....	45
Tabla 25: RF-013.....	45
Tabla 26: RF-014 .....	45
Tabla 27: RF-015.....	46
Tabla 28: RF-016 .....	46
Tabla 29: RF-017 .....	46



Tabla 30: RNF-001.....	47
Tabla 31: RNF- 002 .....	47
Tabla 32: RNF-003.....	47
Tabla 33: Plantilla de alteración de estados .....	50
Tabla 34: AC-01 - Movimiento del jugador .....	52
Tabla 35: AC-02 - Rotación de la cámara de realidad virtual .....	52
Tabla 36: AC-03 - Activación de ruta de <i>pathfinding</i> .....	53
Tabla 37: AC-04 - Activación de ítems de cambio de nivel .....	53
Tabla 38: Matriz de trazabilidad casos de uso-requisitos .....	54
Tabla 39: Ordenador de desarrollo .....	55
Tabla 40: Ordenador de pruebas.....	55
Tabla 41: Plantilla de validación de requisitos .....	66
Tabla 42: Tabla de validación para RF-001 .....	67
Tabla 43: Tabla de validación para RF-002 .....	67
Tabla 44: Tabla de validación para RF-003 .....	67
Tabla 45: Tabla de validación para RF-004 .....	67
Tabla 46: Tabla de validación para RF-005 .....	67
Tabla 47: Tabla de validación para RF-006 .....	68
Tabla 48: Tabla de validación para RF-007 .....	68
Tabla 49: Tabla de validación para RF-008.....	68
Tabla 50: Tabla de validación para RF-009 .....	68
Tabla 51: Tabla de validación para RF-010.....	69
Tabla 52: Tabla de validación para RF-011.....	69
Tabla 53: Tabla de validación para RF-012.....	69
Tabla 54: Tabla de validación para RF-013.....	69
Tabla 55: Tabla de validación para RF-014.....	70
Tabla 56: Tabla de validación para RF-015.....	70
Tabla 57: Tabla de validación para RF-016.....	70
Tabla 58: Tabla de validación para RF-017.....	70
Tabla 59: Tabla de validación para RNF-001.....	71
Tabla 60: Tabla de validación para RNF-002 .....	71



Tabla 61: Tabla de validación para RNF-003.....	71
Tabla 62: Gastos de personal .....	78
Tabla 63: Gastos de material amortizable .....	79
Tabla 64: Presupuesto total del proyecto.....	79



## INTRODUCCIÓN

En los últimos años, el campo de la realidad virtual ha avanzado a pasos de gigante, gracias en gran medida al sector de los videojuegos, cada vez más realistas y que buscan la inmersión del jugador en el juego. Géneros del sector tales como el terror, FPS (*First Person Shooter*) o simuladores ganan un plus de realismo y ambientación gracias a la realidad virtual.

Dada la creciente relevancia económica del sector, cada vez son más las empresas que intentan aportar su propia solución de realidad virtual, ofreciendo a desarrolladores la posibilidad de adquirir un kit de desarrollo con el hardware de realidad virtual, con el fin de trabajar en proyectos o aplicaciones que se apoyen en la realidad virtual como una característica básica o adicional para ofrecer al usuario una mayor inmersión.

## OBJETIVO Y MOTIVACIÓN

Al tratarse la realidad virtual de un tema innovador, de gran interés y de mucho futuro por delante, surge este trabajo de fin de grado, que intenta acercar esta característica al ámbito de los paseos o visitas virtuales que muchas veces se realizan de lugares con cierta relevancia para ofrecer al público una visita exhaustiva a los mismos.

En la actualidad, la gran mayoría de estas visitas virtuales constan de un pequeño montaje de fotos, acompañados de una serie de vídeos; mientras que otros ofrecen una panorámica visual de 360º desde varios puntos determinados del lugar.

En el presente trabajo se pretende aportar una solución distinta a las mencionadas en el párrafo anterior, empleando la realidad virtual como tecnología principal, y proporcionando por tanto una alternativa mucho más inmersiva. Para ello, se elaborará un prototipo que permitirá al usuario de la aplicación visitar de un modo interactivo el edificio Sabatini de la Escuela Politécnica Superior.

A su vez, se pretende profundizar en el aprendizaje y uso del motor Unity 3D en aplicaciones de este tipo o similares, como podría ser un videojuego; conociendo el alcance y las limitaciones de dicho software, lo que determina, junto al modelo en 3D del edificio, el resultado del proyecto.

La motivación del autor es, principalmente, adquirir una base de conocimientos sobre la realidad virtual, el proceso completo de desarrollo de aplicaciones o juegos, así como reforzar la capacidad de esfuerzo y superación del alumno al verse inmerso en un proyecto con una larga lista de tareas y problemas por resolver.

Con este trabajo de fin de grado se espera dar el primer paso hacia proyectos y aplicaciones de mayor envergadura a lo largo de la carrera profesional del autor.

## ESTRUCTURA DEL DOCUMENTO

En este apartado se muestra la estructura que presenta el presente documento, buscando detallar la propuesta de una forma ordenada y exhaustiva.

- **Estado del arte:** Comprende un breve resumen de la historia de la realidad virtual que introduce a las características y especificaciones del hardware de VR (*Virtual Reality*) usado en el proyecto. Asimismo, se realiza un estudio de viabilidad de las diferentes opciones que ofrece el mercado para el desarrollo del trabajo, así como una justificación de las alternativas seleccionadas. A su vez, se mencionan algunos trabajos previos, existentes en el archivo de proyectos de la biblioteca, cuya motivación o temática sea similar a la del tratado en este documento.
- **Análisis de la propuesta:** Define los requisitos, funcionales y no funcionales, establecidos para el proyecto, así como un análisis de viabilidad de la propuesta en el marco regulador y el entorno socioeconómico actual.
- **Diseño e implementación de la propuesta:** Aporta el diseño físico-lógico de la aplicación, así como una serie de tareas, en base al diseño y a los requisitos propuestos, a seguir para completar las diferentes partes de las que consta el trabajo.
- **Gestión del proyecto:** Expone la planificación estimada para este proyecto, así como el presupuesto previsto para el completo desarrollo de la aplicación.
- **Validación y evaluación:** Muestra el resultado de las pruebas realizadas con la aplicación, junto a la validación de los requisitos determinados en el apartado de "Análisis de la propuesta".
- **Conclusiones:** Finalmente, se ofrecen una serie de conclusiones acerca del resultado final del proyecto, junto a una validación de los objetivos que se marcaron al comienzo del desarrollo, así como unas líneas futuras de cara a mejorar y perfeccionar la aplicación y la experiencia que ofrece al usuario final.



## GLOSARIO DE TÉRMINOS EMPLEADOS

Para una correcta comprensión del presente documento, se exponen a continuación los términos especiales y acrónimos utilizados, ordenados alfabéticamente:

- **FPS:** Disparos en primera persona (*First Person Shooter*)
- **Ítem de cambio de nivel:** Objeto situado en los diferentes niveles de la aplicación que permite al usuario lanzar el escenario correspondiente a dicho ítem.
- **POI:** Punto de interés (*Point of Interest*)
- **Prefabs:** Objetos predefinidos que pueden ser utilizados en cualquier escena del proyecto.
- **Stakeholder:** Individuo o colectivo interesado o implicado en el desarrollo de un proyecto.
- **VR:** Realidad virtual ( *Virtual Reality*)





## ESTADO DEL ARTE

### RECORRIDO HISTÓRICO DE LA REALIDAD VIRTUAL <sup>[8]</sup>

La principal característica del presente trabajo es la integración de un módulo de realidad virtual en una aplicación de visita/paseo virtual por el edificio Sabatini.

Esta rama de la informática ha adquirido una gran popularidad y relevancia en los últimos años, pero en realidad esta disciplina ha venido desarrollándose desde que en 1844, Charles Wheatstone crease el '*estereoscopio*' (mostrado en la ilustración 1), considerado la base de los visores de realidad virtual. Su funcionamiento es sencillo, consiste en la obtención de dos fotografías prácticamente idénticas, pero diferenciadas por un ligero cambio en la toma de la imagen. Cada ojo observará una imagen, y el cerebro las mezclará en una, creando un efecto tridimensional.



Ilustración 1: Estereoscopio [17]

Años más tarde, en 1891, Louis Ducos patenta el '*anaglifo*' (mostrado en la ilustración 3). Se trata de una imagen estereoscópica en la que, para el ojo derecho, se suprime mediante un filtro el color rojo, y para el ojo izquierdo el verde y el azul. Cabe destacar que a partir de esta patente surgen las gafas 3D (mostradas en la ilustración 2) que se han venido utilizando para fines cinematográficos, por ejemplo.



Ilustración 2: Gafas 3D basadas en el concepto de '*anaglifo*' [18]

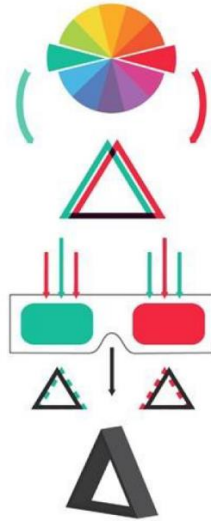


Ilustración 3: Sistema de 'anaglifos' [19]

Ya en el siglo XX, dos empleados de Philco Corp. construyeron el que posiblemente sea el primer casco de realidad virtual que, dotado con sensor magnético que determinaba la posición de la cabeza del usuario, permitía observar imágenes en movimiento.

En 1962 se desarrolló el '*Sensorama*' (mostrado en la ilustración 4), el primer dispositivo que pretendía obtener una inmersión cinematográfica total, mediante imágenes estereoscópicas, sonido estéreo, etc.



Ilustración 4: Cartel promocional de Sensorama [20]

Tres años más tarde surgiría el concepto de realidad virtual de la mano de Ivan Sutherland: *"La pantalla es una ventana a través de la cual uno ve un mundo virtual. El desafío es hacer que ese mundo sea vea real, actúe real, suene real, se sienta real."*

Se fueron sucediendo pequeños avances en esta tecnología hasta que Lawrence G. Roberts y el propio Sutherland diseñaron sendos prototipos para eliminar superficies oscuras y ocultas de una imagen (permitiendo así la utilización de gráficos 3D) y de optimización de la técnica anterior.

A partir de ahí, la realidad virtual tomó el camino de las aplicaciones de carácter militar, generalmente llevándose a cabo en simuladores de vuelo, primero con displays gráficos, y más adelante con la inclusión de cascos de simulación.

Pero los avances en VR no se centraron únicamente en lo visual, sino que se extendieron a otro tipo de percepción.

En 1983, el Dr. Gary Grimes patentó el primer guante que reconocía la posición de la mano (según se aprecia en la ilustración 5), mediante sensores de flexión, táctiles, de orientación y de posición. Su objetivo era reconocer caracteres alfanuméricos en base a la posición de la mano para poder sustituir los teclados convencionales por este guante especial.

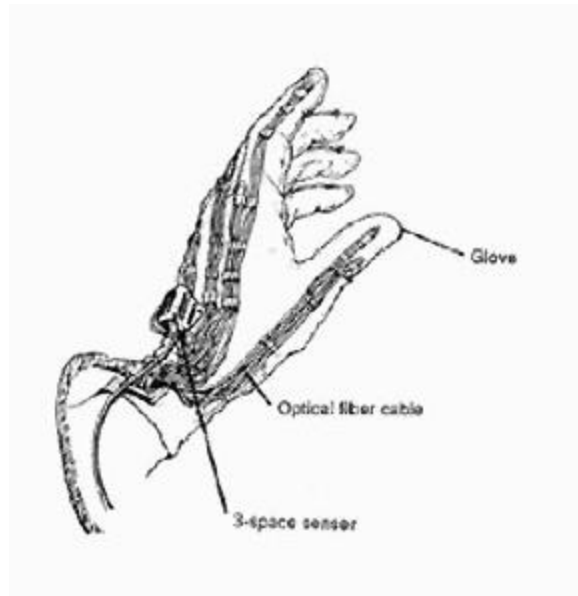


Ilustración 5: Guante de VR de Gary Grimes [21]

Un par de años más tarde, gracias a la colaboración de la NASA, se completaron dos proyectos de realidad virtual, 'VIVED' (Virtual Environment Display System) <sup>[1]</sup> y el casco de VR NASA-Ames <sup>[9]</sup> (mostrados ambos en las ilustraciones 6 y 7, respectivamente). Se trata de hardware de realidad virtual muy avanzado, constituyendo una estación de bajo coste destinada a astronautas de la NASA, dotado con un amplio campo de visión, sonido estéreo y sensores de posición en el casco, así como el primer sistema práctico de visores estereoscópicos; respectivamente.

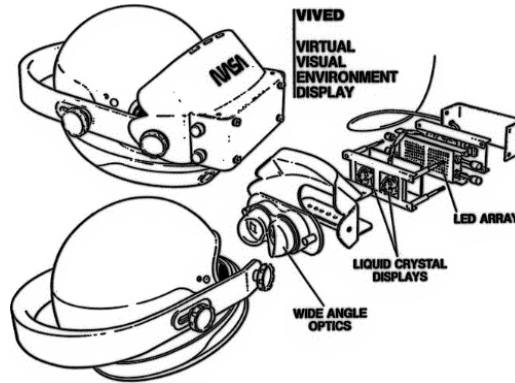


Ilustración 6: 'VIVED' [22]



Ilustración 7: Casco VR NASA-Ames [23]

A partir de este punto, la tecnología siguió avanzando y se empezó a comercializar hardware de realidad virtual, no tan avanzado como fueron los dos ejemplos mostrados anteriormente, pero que sí permitían al usuario percibir de una forma más interactiva e inmersiva las aplicaciones compatibles con esta tecnología.

Desde entonces, donde más se ha dejado querer la realidad virtual ha sido en el sector de los videojuegos<sup>[7]</sup>. Con mayor o menor éxito, consolas y accesorios como *SEGA VR* o la *Virtual Boy* de Nintendo fueron pequeños pasos que el desarrollo de la realidad debía superar para llegar al consumidor en general. Así llegaron éxitos como *Wii*, *Kinect*, o el hardware de realidad virtual más utilizado actualmente, Oculus Rift, que ofrecen al usuario una experiencia de juego superior pero intuitiva a su vez.



Ilustración 8: De izquierda a derecha: Virtual Boy, SEGA VR, Wii, Kinect, Oculus Rift [24][25][26][27][28]

## ESTUDIO DE ALTERNATIVAS

La informática es una disciplina en constante movimiento, y existe una competencia feroz entre la inmensa variedad de multinacionales y grandes empresas que comercializan hardware y software y, en la mayoría de los casos, invierten en I+D para obtener un producto innovador, de calidad y que atraiga al consumidor. Es por eso que, incluso en el campo de la realidad virtual, existen múltiples alternativas entre las que elegir para desarrollar nuestra aplicación y ofrecer esa inmersión que se desea en el presente trabajo.

Asimismo, desde hace bastantes años se vienen ofreciendo y renovando multitud de motores gráficos y SDK para, principalmente, el desarrollo de videojuegos.

La calidad y el número de funcionalidades que incluyen estos motores y kits van en aumento cada año, pudiendo, con el equipo de programadores y diseñadores adecuado, crear verdaderas obras de arte (como se puede observar en la ilustración 9).



Ilustración 9: Captura de Bioshock Infinite [29]

A continuación se exponen las diferentes opciones que se han barajado para el motor gráfico y SDK, así como para el hardware de realidad virtual.

Puesto que se disponía de un modelo básico del Sabatini, modelado en Blender, del que se podían reutilizar algunos elementos, se ha optado por continuar haciendo uso de esta herramienta de modelado, por lo que no se considera necesario estudiar otras alternativas en este sentido.

## HARDWARE DE REALIDAD VIRTUAL

Este estudio expone las múltiples opciones de hardware de realidad virtual existente en la actualidad, bien se trate de proyectos en fase de desarrollo o hardware de plataforma única, con el fin de evaluar el abanico de alternativas y escoger la que resulte más oportuna para el desarrollo de la propuesta.

Hoy en día, las opciones de VR que podemos encontrar son las siguientes:

---

### PROJECT MORPHEUS

La propuesta de Oculus para la realidad virtual es un proyecto muy similar al que ofrece Sony con su Project Morpheus <sup>[2]</sup> <sup>[11]</sup> <sup>[12]</sup>, ofreciendo unas especificaciones técnicas parecidas al DK2 de Oculus, que se muestran en la tabla 1.

El modelo se caracteriza por la inclusión de tecnología de audio 3D, que recrea sonidos estereoscópicos en todas direcciones, adaptándose en tiempo real en función de la posición de la cabeza del usuario; además de ser 100% compatible con los controladores de PlayStation 4, DualShock 4 y PS Move, lo que aumenta en gran medida la experiencia de realidad virtual.

Como se puede observar, Sony apuesta por la realidad virtual exclusivamente para su PlayStation 4, sin abrirse a un mercado más variado como sí hacen, por ejemplo, Oculus o Avegant.



Ilustración 10: Project Morpheus [30]

Como se puede apreciar, el diseño es más futurista que el que ofrece el resto de alternativas, muy acorde al diseño de PlayStation 4.

Dado el avanzado estado de desarrollo de Morpheus no se descarta que, junto al DK2 de Oculus, sea de las primeras en llegar al salón del consumidor, una vez que finalicen el desarrollo y lancen la primera versión pública.

## GLYPH

Desarrollado por Avegant, se trata de una propuesta de realidad virtual cuyas características son únicas respecto a sus competidoras, que consiste en lo que denominan '*virtual retinal display*', que utiliza óptica para proyectar imágenes a 720p en movimiento directamente en los ojos del usuario, como si de un televisor de 80 pulgadas se tratase <sup>[3]</sup> <sup>[4]</sup>.

El dispositivo se compone del módulo óptico, junto a unos auriculares de alta calidad y cancelación de ruido, que incrementan en gran medida la experiencia del usuario. La calidad de la imagen, a su vez, es vívida, con colores brillantes, a diferencia de lo que ocurre con la mayoría de los displays pequeños.



Ilustración 11: Avegant Glyph <sup>[31]</sup>

Glyph es compatible con gran cantidad de juegos existentes en el mercado, pero con un pequeño inconveniente, la pantalla visible aparece con bordes negros, al no tratarse de los 360° que se pretenden en un mundo de realidad virtual.

Posee un campo de visión de 40°, muy inferior al del resto de alternativas, pero tratándose de un dispositivo con unas características tan especiales, y orientado, de momento, hacia la visualización de contenido multimedia, esta desventaja resulta atenuada.

A continuación se presenta un breve resumen del funcionamiento de la óptica de este dispositivo:



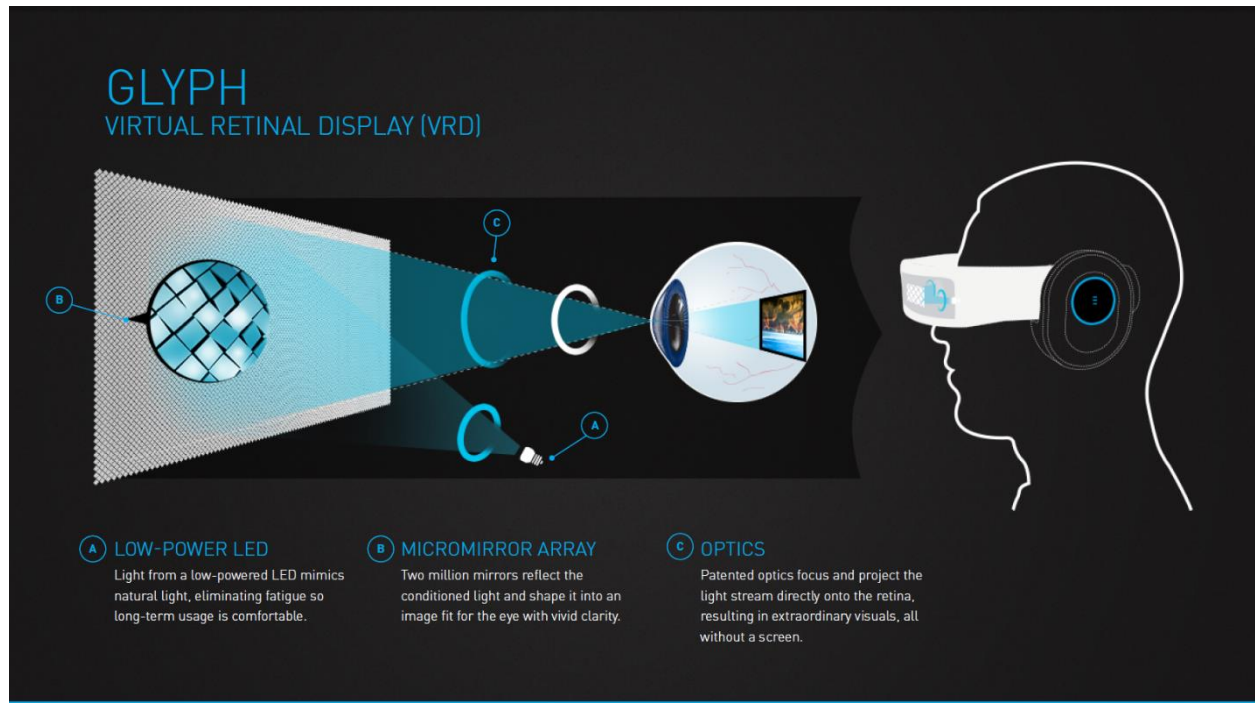


Ilustración 12: Avegant Glyph Virtual Retina Display [32]

La tecnología 'Virtual Retinal Display' que utiliza Glyph consiste, según la imagen anterior, en el uso de un millón de microespejos por cada ojo para reflejar una imagen clara, realista y vívida directamente en la parte posterior de la retina. Se trata de un proceso avanzado e innovador que resulta en una experiencia única que, según usuarios que han podido probarlo, no provoca ningún tipo de mareo o náusea tras su uso.

Este prometedor dispositivo aparecerá en el mercado, previsiblemente, a finales del presente año.



---

## INFINITEYE

Desarrollado en Francia, este prototipo de hardware de realidad virtual posee unas características muy similares a las del DK1 de Oculus Rift, ofreciendo como característica principal un campo de visión de 210°, inmensamente superior al del resto de alternativas <sup>[10]</sup>. A su vez, la resolución de InfinitEye mejora ligeramente la del DK1 de Oculus, siendo inferior ante el resto de alternativas.

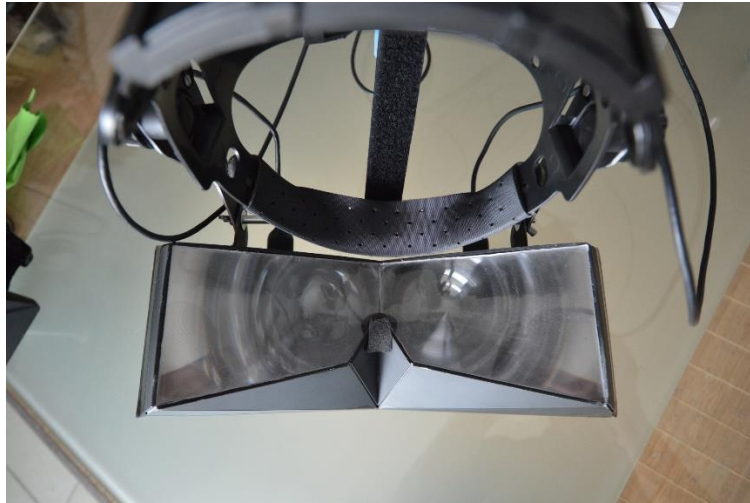


Ilustración 13: InfinitEye [33]

Por desgracia, pese al prometedor avance en campo de visión que aporta este prototipo, son nulas las novedades o actualizaciones al respecto, lo que oscurece en parte el futuro de este proyecto, pese a haber recibido ayuda de colaboradores gracias a la plataforma Kickstarter.

---

## OCULUS RIFT:

La apuesta de Oculus para realidad virtual es el ya clásico y aclamado Oculus Rift.

Actualmente en su segunda versión (DK2), el dispositivo de Oculus ha ido recibiendo gran cantidad de mejoras desde que apareció el primer prototipo en 2012, ofreciendo actualmente unas características muy superiores a las de su primera versión, ofreciendo, en su versión 'Crystal Cove' (Enero de 2014), la opción de 1080p, junto al uso del primer display con tecnología OLED del mercado y un nuevo sistema de detección del movimiento mediante una cámara externa que detecta la posición de unos puntos infrarrojos situados en el dispositivo, lo que permite al sistema detectar acciones del usuario tales como agacharse.



Ilustración 14: Oculus Rift DK1 [34]



Ilustración 15: Oculus Rift DK2 [35]

La tecnología OLED para el display que introduce el DK2 implica un mayor nivel de detalle y una reducción significativa del denominado '*screen door effect*', que consiste en la apreciación de la separación de los píxeles en pantalla. La diferencia entre las versiones de Oculus relacionadas con este efecto puede apreciarse a continuación.

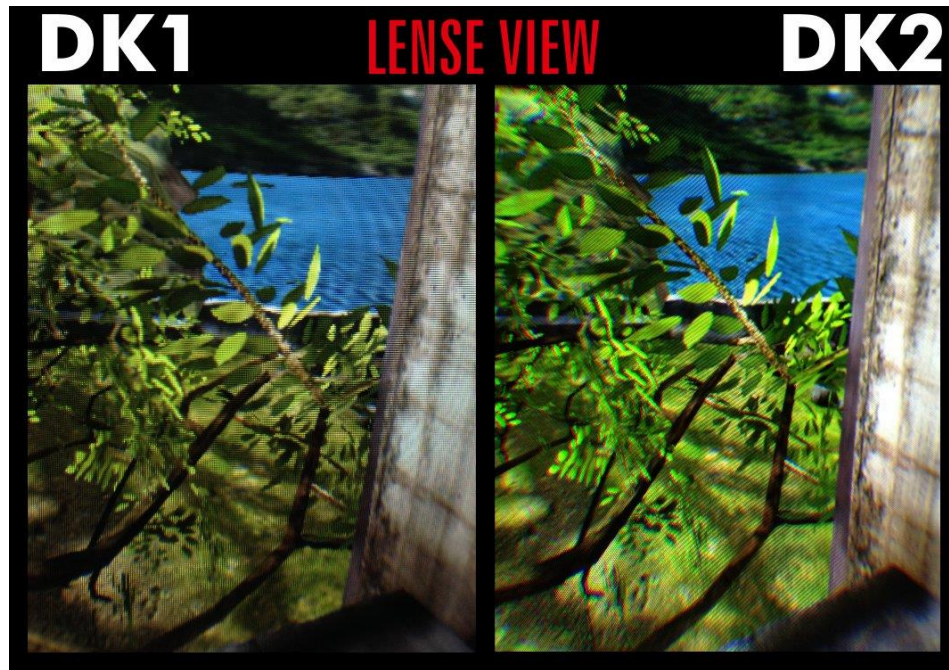


Ilustración 16: Comparación del 'screen door effect' en Rift DK1 y DK2 [36]

Asimismo, el DK2 ofrece una tecnología patentada por Valve, 'baja persistencia de visión', cuyo objetivo es mostrar la información más reciente relativa al movimiento del usuario.

La otra gran ventaja que aporta el DK2 respecto a su predecesor es la inclusión, como se mencionó anteriormente, de un sistema óptico de seguimiento basado en una cámara externa, el cual, junto a la unidad de medición inercial que incorpora el dispositivo y ayudándose de LEDs infrarrojos, proporciona al DK2 la capacidad de medir tanto el movimiento rotacional como el de traslación [5].

Todas estas bondades hacen del DK2 de Oculus Rift el dispositivo de realidad virtual más completo del mercado.

Por último, pero no menos importante, cabe destacar la gran cooperación y feedback que ofrece la comunidad de desarrolladores que trabaja con Rift DK1, que ha ayudado, en parte, al desarrollo de su sucesor.

## COMPARATIVA DE HARDWARE DE REALIDAD VIRTUAL

Una vez exploradas las diferentes opciones de VR, resulta imprescindible decantarse por una de ellas.

A continuación se muestra una tabla con las especificaciones gráficas de las opciones valoradas anteriormente:

Especificaciones del display de VR					
	Morpheus	Glyph	Rift DK1	Rift DK2	InfinitEye
Resolución nativa (X)	1920	1280	1200	1920	1280
Resolución nativa (Y)	1080	720	800	1080	800
Resolución por ojo	960 x 1080	1280 x 720	640 x 800	960 x 1080	640 x 800
Tecnología	LCD	VRD (LED)	LCD	OLED	LCD
Campo de visión	90°	40°	110°	100°	210°
Precio	< \$599	\$499	\$300	\$350	~ \$450

Tabla 1: Especificaciones del display de VR

De la tabla y de los análisis anteriores se pueden extraer las siguientes conclusiones:

- Los mejores dispositivos a nivel de resolución y calidad gráfica son Morpheus, Glyph y Rift DK2.
- Sin embargo, el campo de visión aportado por InfinitEye es muy prometedor, y sería un punto a tener en cuenta si alguna de sus competidoras aumentase su campo de visión para acercarse a InfinitEye.
- Respecto al Rift DK1, pese a tener unas características similares a las de InfinitEye, se podría considerar como obsoleto, al estar su nueva versión disponible.
- Por otra parte, Morpheus, la apuesta de Sony, es de uso exclusivo para PlayStation 4, lo cual hace que, para el desarrollo del presente trabajo, quede automáticamente descartado.
- Glyph es una opción realmente interesante, pero como se menciona anteriormente, es un dispositivo cuyo fuerte está orientado hacia la reproducción multimedia, mientras que este trabajo está dirigido más hacia un simulador o un videojuego, perdiendo la calidad que obtiene en la otra alternativa.
- Ni InfinitEye ni Glyph están disponibles actualmente, lo que, obviamente, impide que sean opciones válidas para el desarrollo del proyecto.
- A su vez, se desconoce la compatibilidad de InfinitEye y Glyph con motores gráficos o SDKs existentes, lo cual dificulta de igual manera su elección.
- Pese a su obsolescencia, el DK1 de Oculus cuenta con el respaldo de una enorme comunidad de desarrolladores, cuyos plugins, manuales, sugerencias y resolución de dudas, que sin duda facilitan la labor de los nuevos desarrolladores que se incorporan al mundo de la realidad virtual.

Debido a las peculiaridades del hardware de realidad virtual, la compatibilidad de éste con motores gráficos y SDKs es imprescindible para desarrollar un proyecto como el presente. Ambas versiones de Oculus Rift son compatibles con Unity 3D, Unreal Engine 4 y CryEngine.

Según las conclusiones extraídas anteriormente, el autor considera que la opción más óptima es el DK2 de Oculus, por el feedback y la comunidad que tiene detrás, que contribuyeron a su desarrollo, así como la compatibilidad con varios motores gráficos y las sólidas características gráficas que ofrece.

A continuación se realizará un estudio de los motores gráficos y SDK compatibles con Oculus Rift DK2.

---

## MOTORES GRÁFICOS Y SDK

De igual forma que en el apartado anterior, el abanico de posibilidades respecto a motores gráficos y SDK de desarrollo de aplicaciones y juegos es realmente amplio, mucho más que en lo relativo a la realidad virtual, como cabe esperar.

En esta comparativa se analizarán los siguientes aspectos de cada opción:

- **Usabilidad:** Se estudiará la interfaz de usuario, así como la curva de aprendizaje necesaria para emprender proyectos con dicho motor.
- **Funcionalidad:** Se estudiarán las diferentes capacidades del motor en cuestión, "lo que puede hacer".
- **Compatibilidad con Oculus Rift DK2:** Se estudiará si el motor en cuestión es compatible con Rift y, en caso de que lo sea, se determinará la dificultad a la hora de integrar el módulo de VR con el proyecto.
- **Precio:** Influye en gran medida a la hora de adquirir un plan de suscripción a la versión profesional, si procede.

Las alternativas escogidas son los motores más conocidos y utilizados hoy en día: Unity 3D, Unreal Engine 4 y CryEngine.

A continuación se muestra el análisis realizado:

## UNITY 3D

Considerada la alternativa más orientada hacia el desarrollo 'indie' de las tres que se analizan, cada vez más estudios de renombre en la industria han adoptado este motor de desarrollo para sus proyectos [6] [16]. Se trata de un motor potente, con una interfaz de usuario amigable e intuitiva, que permite al desarrollador controlar su proyecto de forma eficiente desde el principio. La curva de aprendizaje, por su parte, es inferior a la del resto de alternativas, facilitando la entrada de desarrolladores noveles al sector. Asimismo, Unity ofrece la capacidad de integración multiplataforma para sus proyectos, permitiendo portar proyectos a entornos tan dispares como Android, PlayStation 3 o navegadores web.

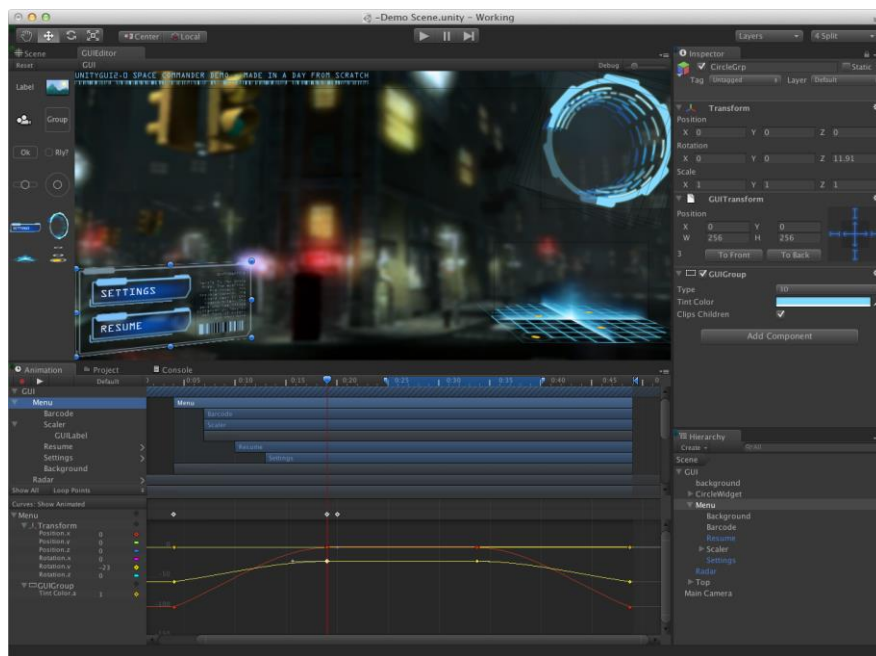


Ilustración 17: Interfaz Unity 3D [37]

Una característica muy importante es que soporta modelos de las aplicaciones de diseño 3D más relevantes, como 3ds Max, Maya o Blender, eliminando así las restricciones a la hora de compatibilizar modelos de diferentes formatos. Además incluye la opción de crear entornos y *sprites* en 2 dimensiones de forma nativa.

Sin embargo, Unity no ofrece las funcionalidades de modelado o construcción de objetos más allá de formas simples como tetraedros, cilindros o planos, por lo que siempre será necesario el uso de una herramienta de terceros para realizar modelos en 3D para el proyecto en cuestión.

Por otra parte, Unity 3D ofrece su versión básica completamente gratis, con la que se pueden realizar proyectos considerables sin la necesidad de pagar. Esto sin duda facilita a desarrolladores con bajo



presupuesto la realización de sus proyectos, lo que ha aumentado significativamente la popularidad y relevancia de este motor de desarrollo.

Finalmente, algo que sin duda caracteriza a Unity respecto a sus competidores es la integración de Oculus Rift en los proyectos. Sólo se necesita un pequeño plugin, descargable desde el portal de desarrolladores de Oculus. El problema es que esta integración, junto a otras funcionalidades como la reproducción de vídeos dentro de la aplicación desarrollada, requiere de la licencia de pago de Unity.

---

## UNREAL ENGINE 4

El motor diseñado por Epic Games es el más utilizado en el sector de los videojuegos por la inmensa mayoría de juegos AAA (superproducciones) del mercado actual <sup>[6]</sup> <sup>[16]</sup>. La calidad de producción con Unreal 4 es realmente alta, pero el problema reside en las especificaciones técnicas que requiere, lo que puede hacer que desarrolladores 'indie' o noveles miren hacia otras propuestas para realizar sus proyectos, ya que generalmente no cuentan con tanto presupuesto como empresas de renombre del sector.



Ilustración 18: Interfaz de Unreal Engine 4 [38]

La interfaz que ofrece Unreal Engine 4 es muy completa e intuitiva a su vez, pero al ofrecer una ingente cantidad de opciones, como la edición a tiempo real, físicas avanzadas y luminosidad hiperrealista, o la producción para consolas de nueva generación, la curva de aprendizaje es realmente elevada, algo que no resulta apropiado para desarrolladores noveles.

Por otra parte, Unreal Engine 4 es compatible con Oculus Rift, pero no se trata de una integración directa como ocurre en el caso de Unity 3D, sino que es preciso instalar el SDK de Oculus y trabajar con él para implementar la vista de realidad virtual en los proyectos desarrollados. Esto constituye una desventaja respecto a la rápida y sencilla integración que ofrece Unity.

Por último, no existe de momento licencia gratuita de UE4, por lo que es imprescindible adquirir la versión completa, mediante una cuota de suscripción mensual de 19 dólares.

## CRYENGINE

CryEngine es un motor de desarrollo extremadamente potente diseñado por la compañía Crytek, que se introdujo por primera vez en el lanzamiento del primer título de la saga de videojuegos *Far Cry*. Se utiliza en el diseño de videojuegos para PC y consolas de nueva generación (PlayStation 4 y Xbox One). La potencia visual que ofrece CryEngine es muy superior a la que ofrece Unity, y siendo similar a la que aporta Unreal Engine 4, utilizando luminosidad y físicas realistas y un sistema avanzado de animaciones. [6] [16]

La interfaz de usuario de CryEngine es compleja, pero ofrece una vasta cantidad de opciones de desarrollo 'in-game', a tiempo real, algo que comparte con Unreal Engine 4 pero que Unity 3D no ofrece.

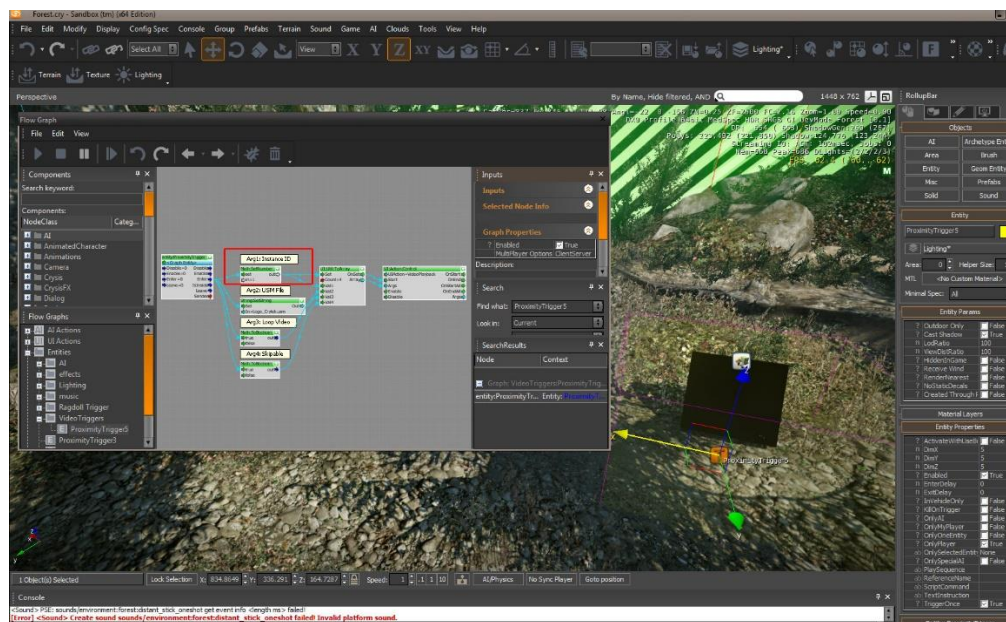


Ilustración 19: Interfaz de CryEngine [39]

Pese a su demostrada potencia en títulos como *Crysis*, la curva de aprendizaje de este motor gráfico es realmente elevada si se pretende utilizar las avanzadas opciones que ofrece, por lo que para un desarrollo





'indie' o que no precise de una calidad de modelos extremadamente detallista es una opción poco apropiada.

Por otra parte, CryEngine ofrece compatibilidad con Oculus Rift, de la misma forma que el resto de opciones pero, al igual que ocurre con Unreal 4, la instalación y configuración del SDK de Oculus para el uso de su hardware de realidad virtual en proyectos desarrollados con CryEngine es algo compleja.

Respecto al precio de suscripción de este motor, la feroz competencia que mantiene con Unreal Engine 4 ha hecho que Crytek ofrezca una suscripción mensual de 10 dólares.

Una vez concluido el análisis de las posibles alternativas, se ha escogido Unity 3D como la más apropiada para el desarrollo de este proyecto, principalmente porque la curva de aprendizaje es mucho menos elevada que la del resto de opciones, algo que beneficia sin duda a desarrolladores noveles. Asimismo, al tratarse de un prototipo, no se considera necesario contar con un realismo impecable o unas físicas avanzadas, lo que sitúa a Unity de nuevo como la mejor alternativa, reduciendo el número de opciones avanzadas que pueden acabar confundiendo al desarrollador. Sin duda la orientación 'indie' de Unity, al disponer de licencia gratuita y de una interfaz de usuario agradable e intuitiva, convierten al motor de desarrollo Unity 3D en la opción elegida.



## ANÁLISIS DE LA PROPUESTA

### ANÁLISIS DE VIABILIDAD DEL SISTEMA

En este apartado se comprobará la viabilidad del proyecto, que abarcará los casos de uso que se establecen para el trabajo, los requisitos definidos para la aplicación (expuestos en el apartado "*Requisitos del sistema*", que se incluye más adelante), la identificación de los *stakeholders* que aparecen involucrados en el proyecto, así como las alternativas que existen a la aplicación a desarrollar y un análisis del marco regulador y del entorno socioeconómico en el que se desenvolverá el proyecto.

---

### IDENTIFICACIÓN DE STAKEHOLDERS

El término *stakeholder* no hace referencia únicamente al usuario final del presente proyecto, sino a cualquier individuo involucrado de alguna forma en el diseño o en cualquiera de las etapas de desarrollo del trabajo y de su implementación.

A continuación se muestran aquellos colectivos o personas que contribuyen en el desarrollo:

- **Usuario final:** Cualquier persona que pudiera utilizar esta aplicación en un futuro. El programa debe ser intuitivo, sencillo de utilizar y atractivo para el público en general, ya que potenciales usuarios podrían ser futuros alumnos de la universidad, por ejemplo.
- **Equipo de desarrollo:** Formado exclusivamente por el alumno (con la colaboración de los tutores), como parte esencial del análisis, diseño, desarrollo e implementación del proyecto.
- **Competencia:** Cualquier sistema que realice una labor similar a la que intenta ofrecer la presente aplicación, que influye en ciertas decisiones de diseño e implementación, permitiendo al equipo de desarrollo buscar el distanciamiento de esta competencia para aportar un producto innovador y atractivo.

---

## ESTUDIO DE LA SITUACIÓN ACTUAL

El objetivo de esta sección es analizar el entorno tecnológico del que dispone el cliente, las alternativas existentes en el mercado que constituyen la competencia para la aplicación que se desarrolla en el presente proyecto.

Este estudio nos permitirá realizar una comparativa entre el sistema a desarrollar y las aplicaciones ya existentes.

---

## VALORACIÓN DEL ESTUDIO DE LA SITUACIÓN ACTUAL

Actualmente existe gran variedad de aplicaciones que pretenden facilitar al usuario una aproximación virtual (gráfica en la mayoría de los casos) a lugares de gran relevancia.

La inmensa mayoría de las aplicaciones que ofrecen algún tipo de visita virtual o aproximación virtual de algún lugar de interés se centran en una sencilla idea: mostrar una serie de fotos clave de 360º, que permiten al usuario rotar una especie de cámara central que simula la visión global que, en este caso, ofrece la aplicación a desarrollar. Estas fotografías vienen acompañadas, normalmente, de un vídeo de presentación del lugar que se visita.

La gran similitud que existe entre las diferentes opciones del mercado hace que el presente trabajo parta con ventaja, al ofrecer la inmersión que aporta la tecnología de realidad virtual, pese a no ser tan realista como las fotografías de alta resolución de las alternativas.

La opción que nos ocupa es, pues, una opción menos realista y detallada que sus competidoras, pero mucho más interactiva e inmersiva, lo que se presupone puede ser más atrayente para el usuario final.

## ANÁLISIS DEL MARCO REGULADOR Y EL ENTORNO SOCIOECONÓMICO ACTUAL

Para el correcto desarrollo e implementación de la aplicación, así como para no incurrir en problemas legales, el proceso completo se acata a la legislación vigente española.

Cabe mencionar que el modelo del edificio Sabatini utilizado en este proyecto es propio del autor, y está basado en un trabajo realizado durante la asignatura Informática Gráfica, correspondiente al curso 2012/2013, por lo que no se trata de un modelo apropiado indebidamente.

El resto del proyecto se realiza con el motor gráfico y el SDK de Unity 3D, de libre distribución, utilizando la versión Pro del mismo para ciertas funcionalidades, incluyendo la compatibilidad con el hardware de realidad virtual Oculus Rift, cuyo kit de desarrollo fue adquirido por el Departamento de Informática de la Universidad.

Se considera, por tanto, que este trabajo de fin de grado se ajusta a la normativa vigente.

Respecto al entorno socioeconómico en el que se desenvuelve el proyecto, el autor se ciñe al presupuesto expuesto en el apartado “*Gestión del proyecto*”.

### DEFINICIÓN DE CASOS DE USO

A la hora de diseñar una aplicación de este calibre, es necesario identificar una serie de casos de uso, para facilitar las labores de especificación de requisitos, así como el diseño de diagramas de clases y de flujo de datos, como se aprecia en el apartado “*Desarrollo e Implementación de la propuesta*”.

La tabla que se muestra a continuación sirve como plantilla para los casos de uso identificados:

Identificador	
Descripción	
Pre-condiciones	
Escenario	
Post-condiciones	

Tabla 2: Plantilla de casos de uso

A continuación se incluye una breve descripción de los campos que componen la tabla:

- **Identificador:** Nombre identificativo del caso de uso. Se utilizará la nomenclatura CU-<XX>, donde CU hace referencia a ‘caso de uso’ y <XX> será un código numérico desde 01 a 99.
- **Descripción:** Indica en qué consiste dicho caso de uso.
- **Pre-condiciones:** Condiciones iniciales que se deben cumplir para realizar el caso de uso.



- **Escenario:** Conjunto de pasos a seguir para realizar el caso de uso.
- **Post-condiciones:** Estado del sistema tras realizarse el caso de uso.

Los casos de uso identificados para la propuesta son los siguientes:

CU-01	
Descripción	Inicio de la aplicación
Pre-condiciones	Disponer del ejecutable de la solución tal y como se distribuye, con todas las dependencias necesarias para su funcionamiento.
Escenario	<ol style="list-style-type: none"><li>1. Obtener el programa</li><li>2. Comprobar que se incluye el ejecutable de Oculus Rift, el propio de la aplicación, y la carpeta de datos.</li><li>3. Iniciar el programa mediante doble click en el ejecutable de Oculus Rift (que activa el módulo de realidad virtual).</li></ol>
Post-condiciones	La aplicación se encuentra iniciada, en el menú principal.

Tabla 3: CU-01 - Inicio de la aplicación

CU-02	
Descripción	Cierre de la aplicación
Pre-condiciones	El programa se encuentra iniciado y en el menú principal.
Escenario	Una vez que el usuario se encuentre en el menú principal, puede cerrar la aplicación pulsando el botón 'Salir' del propio menú.
Post-condiciones	El programa se ha cerrado y el usuario se encuentra de nuevo en su anterior aplicación activa.

Tabla 4: CU-02 – Cierre de la aplicación

CU-03	
Descripción	Primera carga del nivel principal
Pre-condiciones	El programa se encuentra iniciado y en el menú principal.
Escenario	<p>Una vez que el usuario se encuentre en el menú principal, puede realizar la carga inicial del nivel principal pulsando el botón 'Iniciar' del propio menú.</p> <p>Junto a la carga del nivel principal se crea la instancia de jugador, que incluye el módulo de realidad virtual.</p>
Post-condiciones	El sistema realiza la carga inicial del nivel principal, y el usuario ya dispone del controlador de jugador junto al módulo de realidad virtual.

Tabla 5: CU-03 – Primera carga del nivel principal

CU-04	
Descripción	Movimiento del jugador por el nivel actual
Pre-condiciones	La aplicación ha cargado el nivel correspondiente y el jugador se encuentra activo.
Escenario	Una vez que haya cargado el nivel/piso correspondiente, el usuario podrá mover al personaje mediante las flechas de dirección o las teclas 'W', 'A', 'S' y 'D'.
Post-condiciones	El jugador se encuentra en la posición resultado de las teclas pulsadas.

Tabla 6: CU-04 – Movimiento del jugador por el nivel actual

CU-05	
Descripción	Movimiento de la cámara de realidad virtual
Pre-condiciones	La aplicación ha cargado el controlador del personaje o se ha cargado un escenario correspondiente a un POI.
Escenario	<ol style="list-style-type: none"> <li>1. El usuario ha desplazado al personaje hasta un disparador de cambio de escenario, o bien ha cargado por primera vez el nivel principal.</li> <li>2. El usuario activa dicho ítem y se carga el nivel correspondiente al disparador.</li> <li>3. Se activa el controlador de personaje o la cámara de posición fija, en función del nivel.</li> <li>4. El usuario puede mover la cámara de realidad virtual mediante giros de cabeza.</li> </ol>
Post-condiciones	El sistema detecta la posición a la que apunta la cámara de realidad virtual y ofrece la vista correspondiente en función de dicha posición.

Tabla 7: CU-05 – Movimiento de la cámara de realidad virtual

CU-06	
Descripción	Activación de un ítem de cambio de nivel
Pre-condiciones	El controlador de usuario está activo y el jugador se encuentra en cualquiera de los niveles de la aplicación, salvo en un escenario de POI.
Escenario	<ol style="list-style-type: none"> <li>1. El usuario desplaza al personaje hasta un ítem de cambio de nivel.</li> <li>2. Una vez se encuentre sobre dicho elemento, puede pulsar la tecla 'F' para acceder al escenario correspondiente al ítem de cambio de nivel.</li> <li>3. En caso de que se trate de un ítem disparador correspondiente a un POI, el usuario deberá pulsar la tecla 'G', en vez de 'F'.</li> </ol>
Post-condiciones	La aplicación carga el escenario correspondiente.

Tabla 8: CU-06 – Activación de un ítem de cambio de nivel



CU-07	
Descripción	Activación de ruta de pathfinding
Pre-condiciones	El controlador de usuario está activo y se encuentra en una zona mapeada por el módulo de navegación de Unity 3D.
Escenario	<ol style="list-style-type: none"><li>1. El usuario desplaza al personaje hasta la zona mapeada por Unity para la navegación.</li><li>2. El usuario podrá pulsar una de las teclas numéricas 1, 2, 3 ó 4 para activar una de las cuatro rutas definidas para el prototipo.</li></ol>
Post-condiciones	El controlador de personaje se desplaza automáticamente siguiendo la ruta definida.

Tabla 9: CU-07 – Activación de ruta de *pathfinding*

CU-08	
Descripción	Reproducción y controles de escenarios POI
Pre-condiciones	El usuario ha activado un ítem de cambio de nivel y la aplicación ha cargado el nivel POI correspondiente.
Escenario	<ol style="list-style-type: none"><li>1. Una vez cargue el nivel, se deshabilita el controlador de jugador y se inicia el vídeo de presentación de dicho POI.</li><li>2. El usuario puede mover la cámara de realidad virtual de la misma forma que para un nivel normal.</li><li>3. El usuario puede pausar y reanudar, en cada caso, el vídeo actual mediante el uso de la barra espaciadora.</li></ol>
Post-condiciones	El vídeo completa su reproducción.

Tabla 10: CU-08 – Reproducción y controles de escenarios POI

CU-09	
Descripción	Carga de nivel (excepto POI)
Pre-condiciones	El usuario ha activado un ítem de cambio de nivel.
Escenario	<ol style="list-style-type: none"><li>1. La aplicación carga el escenario correspondiente al ítem activado.</li><li>2. El personaje aparece en el punto de <i>spawn</i> (reaparición) relativo al ítem que activó.</li><li>3. Se habilita el controlador de jugador, en caso de que se hallase deshabilitado.</li></ol>
Post-condiciones	El usuario puede desplazarse por el nivel actual.

Tabla 11: CU-09 – Carga de nivel (excepto POI)



## REQUISITOS DEL SISTEMA

En este apartado se incluyen todos los requisitos, funcionales y no funcionales, para el software que se plantea en este trabajo. Dichos requisitos son la base sobre la que se realiza el desarrollo de la aplicación.

Cada uno de los requisitos vendrá definido por una tabla como la siguiente:

Identificador					
Descripción					
Verificabilidad	<input type="checkbox"/> Alta	Necesidad	<input type="checkbox"/> Esencial	Prioridad	<input type="checkbox"/> Alta
	<input type="checkbox"/> Media		<input type="checkbox"/> Deseable		<input type="checkbox"/> Media
	<input type="checkbox"/> Baja		<input type="checkbox"/> Opcional		<input type="checkbox"/> Baja

Tabla 12: Plantilla de requisitos

A continuación se incluye una breve descripción de los campos que componen la tabla:

- **Identificador:** Este campo corresponde al identificador del requisito. Dicho identificador será único e inconfundible, de tal forma que cada requisito sea identificable sin posibilidad de error. Cada uno de los identificadores seguirá la siguiente nomenclatura:
  - **Requisitos funcionales:** RF-<XXX>, siendo <XXX> un número entre 001 y 999.
  - **Requisitos no funcionales:** RNF-<XXX>, siendo <XXX> un número entre 001 y 999.
- **Descripción:** Campo que incluye una descripción del requisito en cuestión.
- **Necesidad:** Este campo indica la necesidad de incorporar el requisito en el sistema. Los posibles valores son:
  - **Esencial:** El requisito debe introducirse obligatoriamente dentro del sistema desarrollado.
  - **Deseable:** La incorporación de estos requisitos en el sistema puede darse en función del desarrollo del mismo.
  - **Opcional:** La incorporación del requisito en el sistema es opcional.

- **Verificabilidad:** Indica la posibilidad de comprobar que el requisito se haya incorporado al sistema. Los posibles valores son:
  - **Alta:** Se puede comprobar de forma sencilla y sin ningún tipo de dudas que el requisito está en el sistema final.
  - **Media:** La comprobación de que el requisito ha sido incorporado en el sistema no tiene una dificultad muy elevada.
  - **Baja:** Es difícil o imposible comprobar que el requisito ha sido introducido en el sistema final.
- **Prioridad:** Este campo indica el grado de prioridad con el que debe ser resuelto un requisito. Los posibles valores son:
  - **Alta:** El diseño e implementación del requisito es de carácter prioritario.
  - **Media:** El requisito se debe diseñar e implementar con prioridad media.
  - **Baja:** El requisito se debe diseñar e implementar con prioridad baja.

## REQUISITOS FUNCIONALES

A continuación se listan todos los requisitos funcionales para la aplicación.

RF-001					
Descripción		Al iniciar la aplicación se mostrará un menú que permita al usuario comenzar la visita virtual o salir de la aplicación			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 13: RF-001

RF-002					
Descripción		La opción 'Comenzar' lanzará el nivel base de la aplicación, creando el controlador del jugador.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 14: RF-002

RF-003					
Descripción		La opción 'Salir' del menú principal cerrará la aplicación.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja

Tabla 15: RF-003

RF-004					
Descripción		El controlador del jugador permitirá al usuario moverse por los niveles utilizando el teclado.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 16: RF-004

RF-005					
Descripción		El controlador del jugador permitirá al usuario mover la cámara gracias a las gafas de VR.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 17: RF-005

RF-006					
Descripción		El jugador podrá desplazarse a los niveles superiores e inferiores, en cada caso, mediante la tecla especificada mientras se encuentra sobre un ítem de cambio de nivel.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 18: RF-006



RF-007					
Descripción		El jugador podrá acceder a las aulas mediante la tecla especificada mientras se encuentra sobre el ítem correspondiente.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 19: RF-007

RF-008					
Descripción		El jugador podrá acceder a los puntos de interés del edificio mediante la tecla especificada mientras se encuentra sobre el ítem correspondiente.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 20: RF-008

RF-009					
Descripción		Cuando se produzca un cambio de nivel, el jugador aparecerá en el punto correspondiente del piso, en función del ítem de cambio de nivel que utilizó.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 21: RF-009

RF-010					
Descripción		En los niveles de puntos de interés, se mostrará un vídeo de corta duración acerca del POI en cuestión.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 22: RF-010



RF-011					
Descripción		Los vídeos mostrados pueden pausarse y reanudarse, en cada caso, mediante el uso de la barra espaciadora.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 23. RF-011

RF-012					
Descripción		Cuando el vídeo mostrado termine, se cargará el nivel correspondiente en la posición anterior del jugador.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 24: RF-012

RF-013					
Descripción		El usuario podrá activar el <i>pathfinding</i> mediante las teclas numéricas '1', '2', '3' y '4', en función de la ruta predefinida que desee tomar.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 25: RF-013

RF-014					
Descripción		La ruta de <i>pathfinding</i> asociada a la tecla '1' llevará al usuario hacia la cafetería.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 26: RF-014



RF-015					
Descripción		La ruta de <i>pathfinding</i> asociada a la tecla '2' llevará al usuario hacia el Punto de Información del Campus.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 27: RF-015

RF-016					
Descripción		La ruta de <i>pathfinding</i> asociada a la tecla '3' llevará al usuario hacia una de las aulas informáticas genéricas del segundo piso.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 28: RF-016

RF-017					
Descripción		La ruta de <i>pathfinding</i> asociada a la tecla '4' llevará al usuario hacia una de las aulas de clase del tercer piso.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 29: RF-017

## REQUISITOS NO FUNCIONALES

A continuación se listan todos los requisitos no funcionales para la aplicación.

RNF-001					
Descripción		La aplicación podrá ejecutarse en varios sistemas operativos además de Windows.			
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional	Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja

Tabla 30: RNF-001

RNF-002					
Descripción		El tiempo de carga entre niveles/pisos deberá ser inferior a un segundo en un ordenador de gama media actual.			
Verificabilidad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 31: RNF-002

RNF-003					
Descripción		La aplicación será capaz de mantener un rendimiento óptimo en ordenadores de variadas características.			
Verificabilidad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja

Tabla 32: RNF-003





## DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA

En este apartado se ofrece el diseño físico-lógico del proyecto, así como una serie de pequeñas secciones donde se profundizará en las diferentes tareas de implementación realizadas a lo largo del desarrollo del trabajo.

### DISEÑO LÓGICO

En este apartado se procede a profundizar en el diseño lógico propuesto. Para ello, y partiendo de la base que conforman los casos de uso y los requisitos definidos, se aportará la siguiente información:

- **Diagrama de clases:** Establece las clases y los atributos de las primeras que componen la aplicación, así como las relaciones entre las propias clases del sistema.
- **Flujo de la aplicación mediante estados:** Detalla la dinámica de la aplicación en base a la relación entre estados.
- **Matriz de trazabilidad de la relación Casos de uso-Requisitos:** Define la relación entre los casos de uso y los requisitos establecidos, lo que se traduce en qué requisitos comprenden qué casos de uso.

---

### DIAGRAMAS DE CLASES

Dado el gran tamaño del diagrama de clases definido para la aplicación, se ha decidido segmentarlo en varias partes e incluirlas como documento adjunto para poder apreciar correctamente los detalles del mismo. Se encuentra en el “*Anexo II: Diagramas de clases*”, al final del presente documento.

La división realizada sobre el diagrama original sigue la lógica a continuación:

- La ilustración 26 de dicho anexo comprende la relación entre el objeto ‘*Master*’, que contiene la lógica del jugador, y los niveles de la aplicación, y define el nivel inicial de la aplicación y los accesos que el usuario tiene al resto de niveles en función de la escena en la que se encuentra. El contenido de las clases correspondientes a los niveles se detalla en las figuras 27 a 30.
- La ilustración 27 de dicho anexo comprende la relación entre el nivel exterior ‘*Outerscene*’ y los niveles POI, ambos accesibles desde dicho nivel. Asimismo, detalla el contenido de los niveles presentes en dicha ilustración.
- La ilustración 28 de dicho anexo detalla el contenido del nivel ‘*piso1*’.

- La ilustración 29 de dicho anexo comprende la relación entre los niveles 'piso2' y 'aulainformatica', ambos accesibles entre ellos. Asimismo, detalla el contenido de los niveles presentes en dicha ilustración.
- La ilustración 30 de dicho anexo comprende la relación entre los niveles 'piso3' y 'aulaclase', ambos accesibles entre ellos. Asimismo, detalla el contenido de los niveles presentes en dicha ilustración.

Se trata, por tanto, de un análisis que parte desde la jerarquía superior de clases de la aplicación, donde se observa al objeto jugador y las diferentes localizaciones a las que puede acceder, y después profundiza en estas localizaciones, detallando los objetos, scripts y variables que forman cada uno de los niveles.

## FLUJO DE LA APLICACIÓN MEDIANTE ESTADOS

Para explicar la dinámica de la aplicación se utilizará un modelo de estados. Se tratará un estado como la posición y orientación del objeto jugador sobre un nivel en concreto en un determinado instante. Se identifica pues este estado por los siguientes aspectos:

- **Posición del objeto jugador en el nivel:** Viene dada por las coordenadas actuales del objeto en los ejes X, Y, Z; junto a la variable *currentFloor*, que indica el piso en el que se encuentra el usuario.
- **Orientación de la cámara:** Viene dada por los valores de los ángulos de Euler (*yaw*, *pitch*, *roll*), que toma la cámara de realidad virtual en función de las rotaciones de cabeza del usuario.

Una vez definido el concepto de estado para el presente proyecto, se establecen las siguientes acciones que alteran dicho estado mediante tablas determinadas en base a la siguiente:

Identificador	
Descripción	
Actúa sobre	
Pre-condiciones	
Post-condiciones	

Tabla 33: Plantilla de alteración de estados

Donde:

- **Identificador:** Nombre identificativo de la acción. Se utilizará la nomenclatura AC-<XX>, donde AC hace referencia a 'acción' y <XX> será un código numérico desde 01 a 99.
- **Descripción:** Indica en qué consiste dicha acción.
- **Actúa sobre:** Indica si la acción en concreto influye en la posición o en la orientación del jugador.
- **Pre-condiciones:** Indica lo que debe hacer el usuario para realizar la acción.
- **Post-condiciones:** Expone las consecuencias de la realización de dicha acción.

Se muestran a continuación las posibles acciones sobre el jugador:



AC-01	
Descripción	Movimiento del jugador
Actúa sobre	Posición del objeto jugador
Pre-condiciones	<p>El usuario puede pulsar las siguientes teclas para mover al jugador:</p> <ul style="list-style-type: none"> <li>• 'W'</li> <li>• 'A'</li> <li>• 'S'</li> <li>• 'D'</li> <li>• Tecla de dirección 'arriba'</li> <li>• Tecla de dirección 'abajo'</li> <li>• Tecla de dirección 'izquierda'</li> <li>• Tecla de dirección 'derecha'</li> </ul>
Post-condiciones	<p>La posición del jugador se ve alterada en función de la tecla pulsada y la duración de la pulsación:</p> <ul style="list-style-type: none"> <li>• 'W' y 'arriba' desplazan al jugador hacia delante, en base a la orientación del mismo.</li> <li>• 'A' e 'izquierda' desplazan al jugador lateralmente hacia la izquierda, en base a la orientación del mismo.</li> <li>• 'S' y 'abajo' desplazan al jugador hacia detrás, en base a la orientación del mismo.</li> <li>• 'D' y 'derecha' desplazan al jugador lateralmente hacia la derecha, en base a la orientación del jugador.</li> </ul> <p>Una pulsación prolongada de cualquiera de las teclas mostradas aumenta la distancia recorrida por el jugador en la dirección correspondiente.</p>

Tabla 34: AC-01 - Movimiento del jugador

AC-02	
Descripción	Rotación de la cámara de realidad virtual
Actúa sobre	Orientación del objeto jugador
Pre-condiciones	Con el hardware de VR debidamente conectado, el usuario puede rotar su cabeza para mover la cámara.
Post-condiciones	La orientación de la cámara y, por tanto, del jugador, se modifica en base a la posición de la cabeza del usuario

Tabla 35: AC-02 - Rotación de la cámara de realidad virtual

AC-03	
Descripción	Activación de ruta de <i>pathfinding</i>
Actúa sobre	Posición del objeto jugador
Pre-condiciones	<p>Mientras el jugador se halle sobre la zona mapeada por la navegación de Unity, el usuario puede pulsar las siguientes teclas numéricas para activar las diferentes rutas de búsqueda disponibles:</p> <ol style="list-style-type: none"> <li>1. Activa la ruta que lleva al usuario a la cafetería.</li> <li>2. Activa la ruta que lleva al usuario al PIC.</li> <li>3. Activa la ruta que lleva al usuario al aula informática del segundo piso.</li> <li>4. Activa la ruta que lleva al usuario al aula de clases del tercer piso.</li> </ol>
Post-condiciones	En función de la ruta activada, el sistema de navegación de Unity calcula la ruta al destino y el objeto jugador se desplaza automáticamente.

Tabla 36: AC-03 - Activación de ruta de *pathfinding*

AC-04	
Descripción	Activación de ítems de cambio de nivel
Actúa sobre	Posición del objeto jugador
Pre-condiciones	<p>Una vez que el jugador se encuentra sobre un ítem de cambio de nivel en concreto, puede pulsar las siguientes teclas para acceder al nivel correspondiente:</p> <ul style="list-style-type: none"> <li>• 'F' permite al usuario acceder al nivel superior siguiente al actual, acceder al nivel inferior siguiente al actual, acceder al nivel de aula correspondiente o, desde un nivel de aula, volver al nivel anterior.</li> <li>• 'G' permite al usuario acceder a un nivel POI.</li> </ul>
Post-condiciones	El objeto jugador se desplaza al nivel activado, y las coordenadas del jugador en este caso se establecen en función del nivel accedido, ya que el jugador aparece en el punto de <i>spawn</i> definido.

Tabla 37: AC-04 - Activación de ítems de cambio de nivel

## MATRIZ DE TRAZABILIDAD DE LA RELACIÓN CASOS DE USO-REQUISITOS

La tabla a continuación define la relación entre los casos de uso identificados para la aplicación y los requisitos funcionales establecidos. De esta forma se observa qué requisitos cubren cada caso de uso, en una relación 'N a N', pudiendo darse los casos de que un caso de uso esté cubierto por varios requisitos y de que un requisito pueda cubrir varios casos de uso a la vez.

		CASOS DE USO								
		CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09
REQUISITOS FUNCIONALES	RF-001	x								
	RF-002			x						
	RF-003		x							
	RF-004				x					
	RF-005					x				
	RF-006						x			x
	RF-007						x			x
	RF-008						x			x
	RF-009						x			x
	RF-010								x	
	RF-011								x	
	RF-012								x	
	RF-013							x		
	RF-014							x		
	RF-015							x		
	RF-016							x		
	RF-017							x		

Tabla 38: Matriz de trazabilidad casos de uso-requisitos

## DISEÑO FÍSICO

En esta sección se expone el diseño físico de la propuesta, en el que se especifican las características de los componentes del sistema requeridos para poner en práctica lo mostrado en el diseño lógico.

Dado que se trata de una visita virtual, muchas de las características de un sistema o aplicación más general, tales como bases de datos o componentes de telecomunicaciones, no son aplicables en este caso.

Los únicos elementos requeridos para las labores de desarrollo, pruebas y, finalmente, ejecución del usuario final, serán los siguientes:

- Ordenador de sobremesa completo u ordenador portátil, en cualquier caso, de gama media actual; lo que asegura un funcionamiento correcto de la aplicación en su estado final.
- Ordenador de sobremesa completo u ordenador portátil, en cualquier caso, de gama alta actual, que será utilizado en labores de desarrollo a lo largo de todo el proyecto, debido a la necesidad de realizar grandes operaciones de cómputo para el diseño del modelo 3D y los componentes desarrollados con el motor gráfico.
- Kit de desarrollo para Oculus Rift (primera versión, DK1), que incluye las gafas de realidad virtual y la conexión de las mismas al ordenador.

Puesto que ya se conocen las especificaciones técnicas del hardware de realidad virtual, a continuación se profundiza en las características del resto de componentes.

Ordenador de desarrollo	
Tipo	Sobremesa
Procesador	Intel Core i5-4670K @ 3.40 GHz
Memoria RAM	8 GB
Disco duro	1 TB
Tarjeta gráfica	NVIDIA GeForce GTX 770
Sistema operativo	Windows 8 Pro 64 bits

Tabla 39: Ordenador de desarrollo

Ordenador de pruebas	
Tipo	Portátil
Procesador	Intel Core i5-M480 @ 2.67 GHz
Memoria RAM	4 GB
Disco duro	640 GB
Tarjeta gráfica	ATI Radeon Mobility HD 5145
Sistema operativo	Windows 7 Pro 64 bits

Tabla 40: Ordenador de pruebas

Junto a estas especificaciones, se hace referencia al documento adjunto “*Anexo I: Guía rápida de configuración del sistema*”, que muestra al usuario los pasos a seguir para preparar el entorno y ejecutar la aplicación una vez dispone de los elementos necesarios.

## FASES DE IMPLEMENTACIÓN

A lo largo del desarrollo completo del presente proyecto, se realizarán una serie de tareas, dependientes entre sí, que llevarán al estado final del prototipo.

Se trata, pues, de las siguientes fases:

---

### FASE DE MODELADO DEL EDIFICIO SABATINI

En primer lugar, es imprescindible contar con los escenarios a través de los cuales se moverá el usuario a la hora de realizar esta visita virtual. Para ello contamos con la base del modelo del Sabatini que se menciona en el apartado “*Estudio de alternativas*”.

De los modelos a realizar se diferencian 4 niveles, como sigue:

- El modelo exterior, que comprende la zona del patio y los soportales de la planta baja. En este nivel se sitúan los accesos a la cafetería y al PIC, los dos POI que se desarrollan en este proyecto.
- El modelo de piso básico, cuyas características permiten que sea replicado y ligeramente modificado para obtener cada uno de los tres pisos superiores de los que dispone el edificio Sabatini.
- El modelo de aula informática, correspondiente a las aulas informáticas del segundo piso.
- El modelo de aula de clase, correspondiente a las aulas grandes del tercer piso.

A continuación se explica el desarrollo de cada uno de estos niveles.

---

### MODELO EXTERIOR

Como se menciona anteriormente, este modelo corresponde a la parte exterior del edificio, incluyendo los accesos a cafetería y PIC.

Según se menciona en el apartado “*Estudio de alternativas*”, se parte de un modelo muy básico del edificio, que se modificará para el propósito de este proyecto, ofrecer la mayor inmersión posible con el mayor realismo que se pueda conseguir dentro de los límites del hardware y el software que se utilizan en el desarrollo.

En el estado inicial, el modelo carecía de cualquier parte interior del edificio, especialmente (siguiendo el diseño de múltiples niveles planteado) de la vista dentro de los soportales, apreciando columnas huecas u omitiendo puertas y ventanas.



La labor es, pues, añadir estos pequeños elementos que faltan para ofrecer una vista exterior lo más completa posible. Para ello se suprimen las paredes básicas que existían en el modelo, y se sacan poco a poco caras a partir de las aristas mediante la herramienta '*Extrude*' que ofrece Blender, la cual permite seleccionar un vértice, arista o cara y, creando una copia del elemento seleccionado, unida a su vez a la original mediante aristas, mover la copia, generando así una nueva figura. Por ejemplo, '*Extrude*' a un vértice generará un vértice y una arista que los conecte, en una arista creará 3 aristas adicionales y se formará una cara, y sobre una cara se formará un cubo.

Así se pueden crear paredes con cualquier tipo de aberturas o incluyendo huecos para situar puertas y ventanas, como se puede observar en los soportales de la planta baja, siempre teniendo en cuenta las medidas y proporciones del edificio y sus elementos.

Realmente, una vez incluidos esos pequeños detalles mencionados anteriormente, no resulta imprescindible realizar muchos más ajustes, puesto que un mayor nivel de detalle y realismo puede incurrir en pérdidas de rendimiento, como caídas de *frames* por segundo, influyendo negativamente en la experiencia del usuario, algo que se pretende evitar en este proyecto.

---

## MODELO DE PISO BÁSICO

En este caso se parte desde cero, ya que el sistema de división en niveles obliga a generar un nuevo modelo que sirva como base para las tres plantas superiores del edificio. Sin embargo, no se trata de volver a crear todas las ventanas, ya que es posible reciclar parte del modelo general del edificio para obtener una planta seleccionando todas las ventanas de una planta mediante la selección en área de Blender, y moviendo todo el *array* de ventanas para poder borrar el resto del edificio, y mantener así las ventanas del piso con las dimensiones del modelo global.

Una vez se dispone de dicho *array*, mediante la creación de planes y la herramienta *Extrude* mencionada en el apartado anterior, es posible generar el resto de la planta, como se puede apreciar en el prototipo que muestra la ilustración 16.

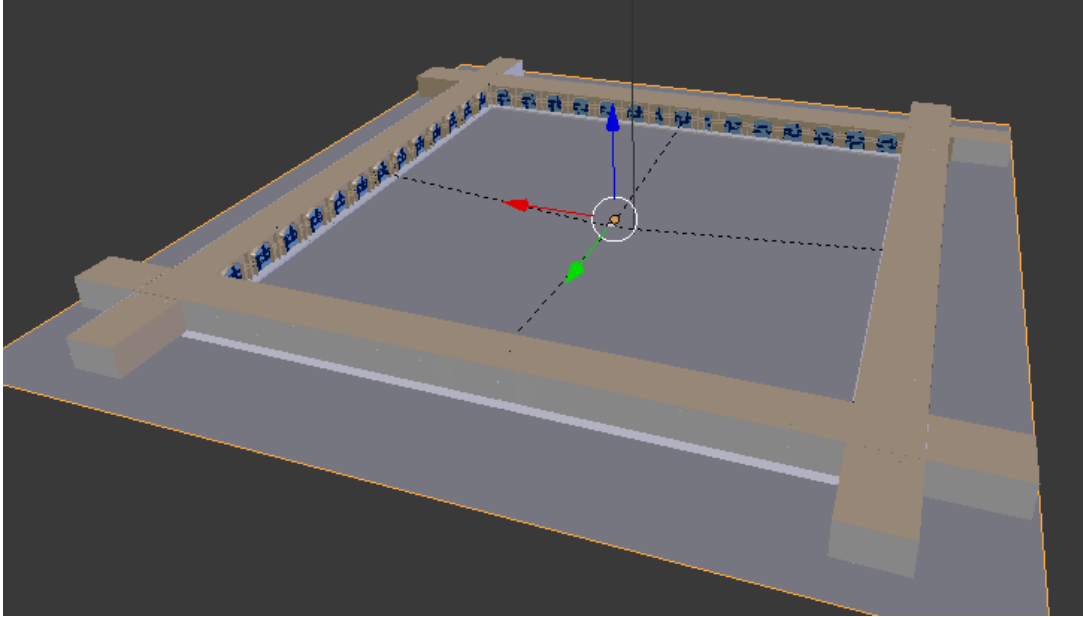


Ilustración 20: Prototipo de piso

A continuación se van añadiendo el resto de pequeños pero relevantes detalles, como pueden ser las puertas de las aulas/despachos o los huecos existentes en el piso.

---

## MODELO DE AULA INFORMÁTICA

De nuevo es necesario modelar desde cero, pero en esta ocasión no es posible reciclar elementos de los dos modelos mencionados anteriormente, ya que no comparten ninguna característica entre sí.

Por tanto, creando la base del aula a partir de un plano, se determina la posición de las puertas, ventanas, columna trasera y pizarra, elementos que pueden ser incluidos sin mayor impedimento.

Una vez aplicadas texturas, el aula queda de la siguiente forma, según muestra la siguiente ilustración:

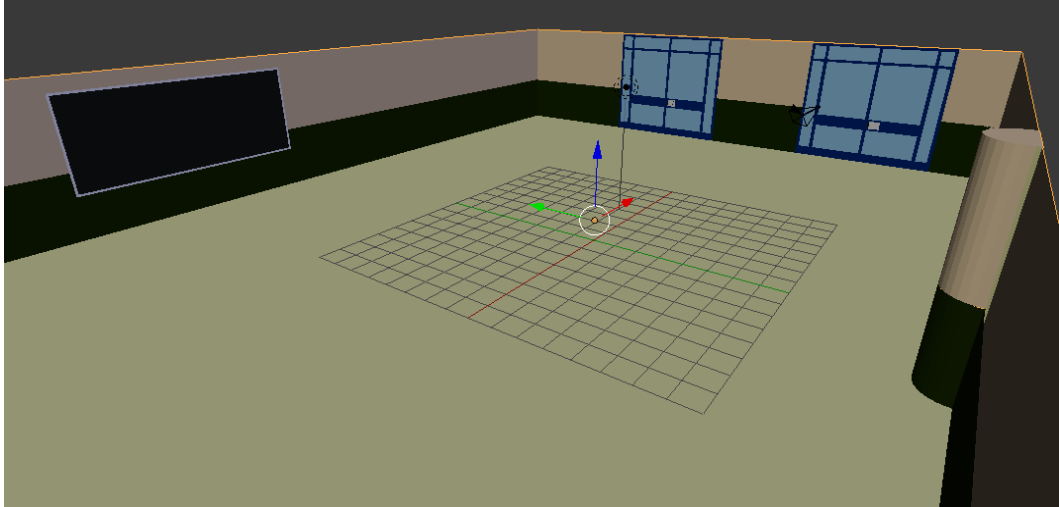


Ilustración 21: Prototipo de aula informática

Por el contrario, para incluir la parte esencial del aula informática, los puestos con ordenadores, se ha recurrido al uso de *assets* (modelos prefabricados) públicos, ya que todos los componentes del puesto de trabajo requieren un laborioso trabajo y una elevada experiencia en modelado gráfico.

Una vez se dispone de todos los elementos del puesto (mesa, sillas de oficina, CPU, monitor, teclado y ratón), lo más lógico es utilizar estos objetos para crear un puesto y, uniendo todos estos elementos en un único objeto, replicarlo para distribuir los puestos por el aula, tal como se muestra en la siguiente figura:



Ilustración 22: Aula informática

## MODELO DE AULA DE CLASE

Por último, es necesario crear un modelo desde el principio para el aula de clase de la tercera planta.

Partiendo de un plano y creando las paredes, se generan una pequeña tarima en base a un cubo y las pizarras características de estas aulas gracias a unos planos y unos cilindros (como se muestra en la ilustración 19). Después se incluyen las dos puertas de las que dispone el aula, junto a las correspondientes ventanas.

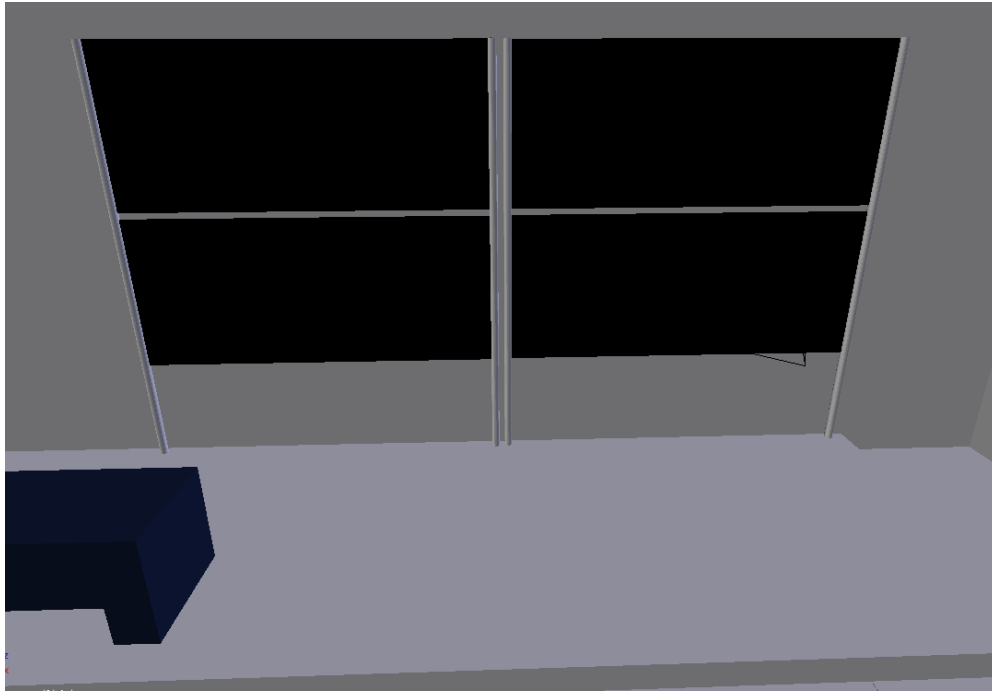


Ilustración 23: Tarima y pizarras del aula de clase

Para crear el techo curvo característico de estas aulas, partiendo de un círculo, se utiliza la herramienta *Knife* de Blender para dividirlo a la mitad y, con el semicírculo obtenido, se aumenta su tamaño proporcionalmente mediante la opción *Scale* para que el diámetro se corresponda con la longitud de la pared frontal. Una vez hecho esto, se utiliza *Extrude* para completar el techo. Por último, se suprime la cara correspondiente al diámetro y el techo ya aparece integrado con el resto del aula, según se observa en la siguiente ilustración:

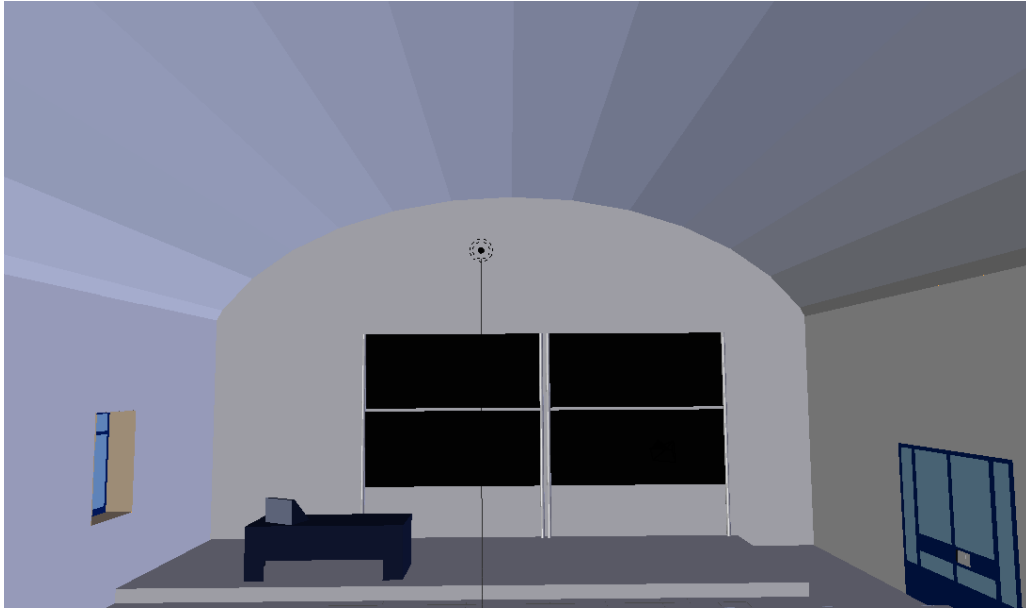


Ilustración 24: Techo curvo del aula de clases

Por último, y como parte más tediosa, se procede a modelar las mesas y los asientos para los alumnos.

A partir de un cubo, el cual se comprime y se estira, curvando los bordes para darle mayor realismo, e incluyendo un *array* de pequeños cubos modificados y replicados, que simulan asientos y respaldos, se forma una fila de asientos para los alumnos. En la imagen a continuación se muestran los puestos para los alumnos:

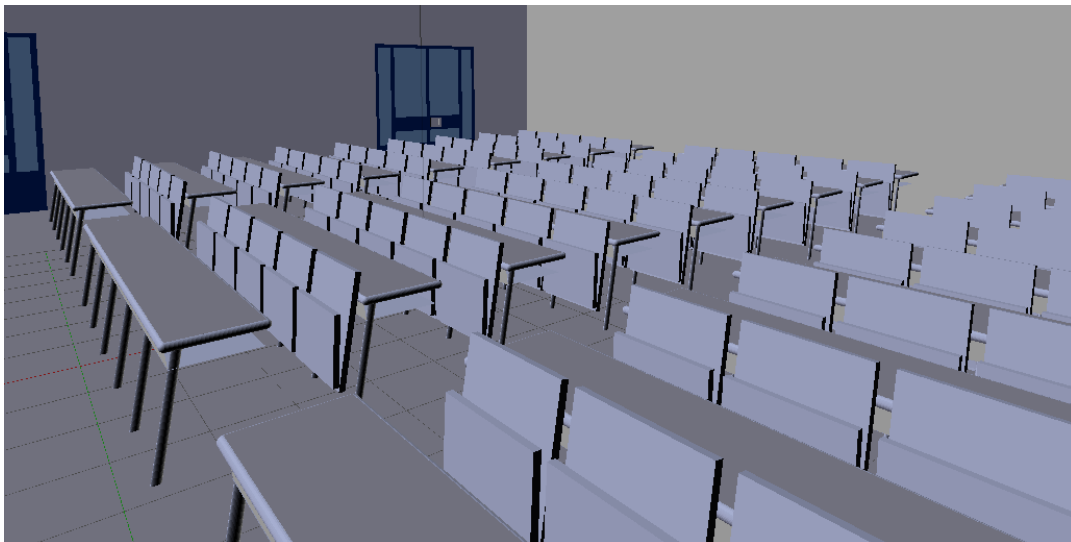


Ilustración 25: Asientos del aula de clases

---

## FASE DE DESARROLLO DE LA APLICACIÓN EN UNITY 3D

En este apartado se profundiza en los aspectos de desarrollo de la aplicación una vez que los modelos de los niveles están preparados, consistiendo en la creación de los niveles, los objetos y scripts necesarios para que el funcionamiento sea el apropiado.

Al igual que en el apartado anterior, se segmentará este bloque en varias partes, para que la información aparezca bien organizada y diferenciada.

Antes de profundizar en el desarrollo, es importante mencionar que no se incluirán líneas de código en este apartado, ya que se trata de scripts considerablemente extensos (en su mayoría) y dependientes unos de otros, lo que sobrecargaría el presente documento de texto innecesario.

---

### JUGADOR

El objeto 'Jugador' es la base del desarrollo de la aplicación en esta fase, puesto que es el elemento que permite al usuario moverse por el edificio Sabatini mientras observa a su alrededor utilizando el hardware de VR. Consta de tres elementos:

- El objeto '*Master*' es un objeto vacío que contiene al objeto 'Jugador' y que se sitúa justo encima del mismo, manteniendo las mismas coordenadas que el jugador a medida que se mueve. A su vez contiene un script, '*MasterController.cs*', cuya función es el paso de variables y datos a través de los diferentes niveles de la aplicación. Para simplificar las labores de programación, este script contiene también toda la lógica del jugador relacionada con el cambio de niveles y de teletransporte a puntos de *spawn* o reaparición.
- El objeto '*OVRPlayerController*' es el objeto 'Jugador' que se menciona en este apartado. Consiste en un objeto controlador de jugador que implementa el movimiento del usuario por los niveles. Mencionar que este objeto forma parte de los componentes que ofrece Oculus a los desarrolladores que utilizan Rift para sus aplicaciones.
- El objeto '*OVRCameraController*' es el objeto controlador de la cámara de realidad virtual del usuario. Está contenido en el objeto 'Jugador'.

Es imprescindible mencionar que el objeto '*Master*', que contiene al jugador y a la cámara, está programado para mantenerse nivel tras nivel, sin crear nuevas instancias de jugador. Por eso el '*Master*' únicamente se genera en la primera carga del nivel exterior, ya que no es necesario crearlo por cada nivel.

---

## NIVEL EXTERIOR

El nivel exterior de la aplicación comprende la planta baja del edificio, así como los POI 'Cafetería' y 'PIC'. Utiliza el modelo exterior detallado en el apartado "*Modelo exterior*", el cual se carga en una escena de Unity, donde se trabaja en el resto de elementos del nivel.

Es necesario habilitar dos ítems de cambio de nivel que correspondan a los POI mencionados en el párrafo anterior, ya que ambos se encuentran en la planta baja y, a su vez, hacen falta cuatro ítems para acceder al primer piso, uno por cada torre del edificio Sabatini. Para ello importaremos como *prefabs* (objetos predefinidos que pueden ser utilizados en cualquier escena del proyecto) un par de objetos modelados en Blender que servirán como ítems de cambio de nivel para acceder a las escenas correspondientes. Una vez importados, se colocan en los puntos más óptimos (frente a las puertas de las torres y de cada POI, respectivamente), y se les añade el script '*teleport.cs*', que contiene la lógica de cambio de niveles para los ítems de cambio de nivel.

---

## PISOS SUPERIORES

La información contenida en este apartado puede aplicarse a cualquiera de los tres pisos superiores del edificio, pues son muy similares.

Los niveles de los pisos superiores comprenden los respectivos pisos a los que corresponden, así como los accesos a las aulas de informática y de clase, situadas en el segundo y tercer piso, respectivamente; y utilizan los modelos de piso básico detallados en el apartado "*Modelo de piso básico*", cargándose cada uno en su nivel correspondiente, donde se trabaja en el resto de elementos de la escena.

Tal como ocurre en el apartado anterior, es necesario habilitar una serie de ítems de cambio de nivel para cada una de las escenas, tal como se detalla a continuación:

- La **primera planta** precisa de ocho ítems, cuatro de ellos para acceder al segundo piso y los otros cuatro para volver a la planta baja.
- El **segundo piso** precisa de nueve ítems, cuatro de ellos para subir al tercer piso, otros cuatro para bajar a la primera planta, y el ítem restante es el que permite al usuario acceder al aula informática.
- La **tercera planta** precisa de cinco ítems, cuatro de ellos para bajar al segundo piso, y el ítem restante es el que permite al usuario acceder al aula de clase.

Todos los ítems de cambio de nivel necesarios para cada una de las plantas llevan, al igual que los expuestos en el apartado anterior, el script '*teleport.cs*'.



De igual forma, los ítems se colocarán en los puntos más apropiados del nivel. Por ejemplo, en el primer y el segundo los ítems de acceso a la planta superior se colocarán junto a la puerta derecha de la torre, y los de bajada al nivel inferior junto a la puerta izquierda de la torre.

---

## AULAS

Estos niveles comprenden las aulas informáticas del segundo piso y las de clase de la tercera planta, y utilizan los modelos de aula informática y de clase detallados en los apartados "*Modelo de aula informática*" y "*Modelo de aula de clase*", respectivamente, cargándose cada uno en su nivel correspondiente, donde se trabaja en el resto de elementos de la escena.

Dado que estas escenas son calles sin salida (desde ellas no se accede a ningún nivel no accedido anteriormente), se precisan dos ítems de cambio de nivel para cada aula, situados en las puertas de las mismas, que retornarán al usuario al nivel correspondiente.

Al igual que en los apartados anteriores, los ítems de cambio de nivel de ambas escenas llevan asociado el script '*teleport.cs*'.

---

## NIVELES POI

Estos niveles comprenden los lugares de interés escogidos de la planta baja del edificio Sabatini, la cafetería y el Punto de Información del Campus. En ambos, cuando el usuario accede, se proyecta un vídeo sobre el POI correspondiente. El usuario tiene la capacidad de pausar y reanudar el vídeo, en cada caso, a placer mediante el uso de la barra espaciadora.

Los niveles POI consisten en un nivel vacío salvo por un plano y una cámara de posición fija, la cual tiene activado el módulo de realidad virtual, por lo que el vídeo se ve de igual manera con las Rift ajustadas. El plano posee una característica muy especial que proporciona la versión profesional de Unity 3D, el uso de *MovieTextures* (que vienen a ser texturas de vídeo). Mediante un pequeño script y esta textura especial, pueden proyectarse clips de vídeo en formato MP4 sobre cualquier superficie, sean esferas, cubos, etc.

Asimismo, el plano lleva asociado un script propio, llamado '*Video.cs*', que contiene la lógica de reproducción del vídeo en los niveles POI. A su vez, cuando el usuario accede a un punto de interés, el script del '*Master*' ('*MasterController.cs*') deshabilita los controles convencionales del jugador, de manera que el usuario únicamente puede usar los controles del vídeo. Una vez que el vídeo termina y el usuario retorna al nivel anterior, el script vuelve a activar el controlador del personaje para que el usuario pueda volver a moverse libremente por las escenas.



---

## PATHFINDING

El presente proyecto incluye una funcionalidad que permite al usuario activar un módulo de *pathfinding*, que le lleve hacia las escenas especiales de la aplicación, a saber:

- Nivel POI de la cafetería, cuya ruta está asociada a la tecla numérica '1'.
- Nivel POI del PIC, cuya ruta está asociada a la tecla numérica '2'.
- Nivel de aula informática, cuya ruta está asociada a la tecla numérica '3'.
- Nivel de aula de clases, cuya ruta está asociada a la tecla numérica '4'.

Esta característica de la aplicación resulta un tanto complicada de implementar, pues depende en gran medida del modelo 3D del lugar en el que se quiere activar la búsqueda.

Para incluir esta funcionalidad en el proyecto se ha utilizado el modo de navegación que ofrece Unity 3D, que consiste en un sistema de mallas de navegación (*NavMesh*) que calcula las superficies sobre las cuales el jugador puede caminar, y a partir del mapeo realizado, el sistema calcula las rutas para llegar hasta las posiciones definidas por un script, la posición de un objeto o una simple coordenada en la escena. El problema encontrado por el alumno en el sistema de mallas es que Unity no detecta ciertas zonas del nivel sobre las que el jugador puede caminar, y no mapea dichas zonas, lo que implica que las rutas que el sistema pueda calcular no llegarán hasta esas zonas.

Aun así, y pese a no ser al 100% una solución completa, el sistema calcula correctamente las rutas hacia los puntos definidos para este prototipo.

Por otra parte, el código correspondiente a este módulo de *pathfinding* se encuentra en el script '*MasterController.cs*', asociado al objeto '*Master*', lo que permite al usuario activar el módulo de *pathfinding* siempre que el jugador esté activo y sobre la zona mapeada por el sistema de mallas de navegación de Unity.

## VALIDACIÓN Y EVALUACIÓN

Esta sección comprende un análisis de los requisitos estipulados con respecto a la dinámica de la aplicación, junto a evaluación realizada por personas anónimas a través de una sencilla encuesta.

### VALIDACIÓN DE REQUISITOS

Una vez completada la implementación de la aplicación, es preciso realizar un análisis del funcionamiento del sistema respecto a los requisitos establecidos, lo que refuerza las decisiones de diseño e implementación tomadas a lo largo del desarrollo del proyecto. Para ello, se toman en consideración todos los requisitos definidos en el apartado “*Requisitos del sistema*” del presente documento, y se verificará que se cumple dicho requisito a través del uso de la aplicación.

La tabla que se muestra a continuación sirve como plantilla para la validación de requisitos respecto a la implementación:

Requisito	
Descripción	
Prueba realizada	
Validación	

Tabla 41: Plantilla de validación de requisitos

A continuación se incluye una breve descripción de los campos que componen la tabla:

- **Identificador:** Este campo corresponde al identificador del requisito. Dicho identificador será único e inconfundible, de tal forma que cada requisito sea identificable sin posibilidad de error. Cada uno de los identificadores seguirá la siguiente nomenclatura:
  - **Requisitos funcionales:** RF-<XXX>, siendo <XXX> un número entre 001 y 999.
  - **Requisitos no funcionales:** RNF-<XXX>, siendo <XXX> un número entre 001 y 999.
- **Descripción:** Campo que incluye una descripción del requisito en cuestión.
- **Prueba:** Describe las acciones a realizar para comprobar el requisito.
- **Validación:** Determina si, con las pruebas realizadas, se puede afirmar que el requisito en cuestión se cumple durante la ejecución de la aplicación.

A continuación se muestran las tablas de validación para los requisitos de la aplicación.



Requisito	RF-001
Descripción	Al iniciar la aplicación se mostrará un menú que permita al usuario comenzar la visita virtual o salir de la aplicación.
Prueba	Se comprueba que al hacer doble click en el ejecutable de la aplicación, arranca y muestra el menú principal con ambas opciones.
Validación	Correcta

Tabla 42: Tabla de validación para RF-001

Requisito	RF-002
Descripción	La opción 'Comenzar' lanzará el nivel base de la aplicación, creando el controlador del jugador.
Prueba	Se comprueba que al hacer clic en la opción 'Comenzar', aparece el nivel exterior y se carga el personaje, pudiendo controlarlo.
Validación	Correcta

Tabla 43: Tabla de validación para RF-002

Requisito	RF-003
Descripción	La opción 'Salir' del menú principal cerrará la aplicación.
Prueba	Se comprueba que al hacer clic en la opción 'Salir' se cierra la aplicación.
Validación	Correcta

Tabla 44: Tabla de validación para RF-003

Requisito	RF-004
Descripción	El controlador del jugador permitirá al usuario moverse por los niveles utilizando el teclado.
Prueba	Se comprueba que se puede mover al jugador por los diferentes niveles de la aplicación donde el controlador está habilitado, utilizando las flechas de dirección o las teclas 'W', 'A', 'S' y 'D'.
Validación	Correcta

Tabla 45: Tabla de validación para RF-004

Requisito	RF-005
Descripción	El controlador del jugador permitirá al usuario mover la cámara gracias a las gafas de VR.
Prueba	Se comprueba que, con las gafas de realidad virtual correctamente conectadas, el usuario puede controlar la cámara mediante movimientos de su cabeza.
Validación	Correcta

Tabla 46: Tabla de validación para RF-005



Requisito	RF-006
Descripción	El jugador podrá desplazarse a los niveles superiores e inferiores, en cada caso, mediante la tecla especificada mientras se encuentra sobre un ítem de cambio de nivel.
Prueba	Se comprueba que se puede acceder a todos los niveles de la aplicación utilizando la tecla 'F' para acceder a niveles superiores e inferiores, en función del piso actual.
Validación	Correcta

Tabla 47: Tabla de validación para RF-006

Requisito	RF-007
Descripción	El jugador podrá acceder a las aulas mediante la tecla especificada mientras se encuentra sobre el ítem correspondiente.
Prueba	Se comprueba que el usuario puede acceder a las aulas de clase e informática mediante el uso de la tecla 'F'.
Validación	Correcta

Tabla 48: Tabla de validación para RF-007

Requisito	RF-008
Descripción	El jugador podrá acceder a los puntos de interés del edificio mediante la tecla especificada mientras se encuentra sobre el ítem correspondiente.
Prueba	Se comprueba que el usuario puede acceder a los niveles de puntos de interés mediante el uso de la tecla 'G'.
Validación	Correcta

Tabla 49: Tabla de validación para RF-008

Requisito	RF-009
Descripción	Cuando se produzca un cambio de nivel, el jugador aparecerá en el punto correspondiente del piso, en función del ítem de cambio de nivel que utilizó.
Prueba	Se comprueba que al activar los ítems de cambio de nivel, el jugador aparece en el punto de reaparición o <i>spawn</i> correspondiente.
Validación	Correcta

Tabla 50: Tabla de validación para RF-009



Requisito	RF-010
Descripción	En los niveles de puntos de interés, se mostrará un vídeo de corta duración acerca del POI en cuestión.
Prueba	Se comprueba que al entrar a un nivel POI, se carga el vídeo correspondiente.
Validación	Correcta

Tabla 51: Tabla de validación para RF-010

Requisito	RF-011
Descripción	Los vídeos mostrados pueden pausarse y reanudarse, en cada caso, mediante el uso de la barra espaciadora.
Prueba	Se comprueba que el usuario puede pausar el vídeo cuando pulsa la barra espaciadora y que puede reanudarlo pulsando de nuevo la misma tecla.
Validación	Correcta

Tabla 52: Tabla de validación para RF-011

Requisito	RF-012
Descripción	Cuando el vídeo mostrado termine, se cargará el nivel correspondiente en la posición anterior del jugador.
Prueba	Se comprueba que, al finalizar el vídeo de cada POI, se carga el nivel correspondiente en la posición determinada.
Validación	Correcta

Tabla 53: Tabla de validación para RF-012

Requisito	RF-013
Descripción	El usuario podrá activar el <i>pathfinding</i> mediante las teclas numéricas '1', '2', '3' y '4', en función de la ruta predefinida que desee tomar.
Prueba	Se comprueba que al pulsar cualquiera de las cuatro teclas asignadas se activa la ruta correspondiente.
Validación	Correcta

Tabla 54: Tabla de validación para RF-013



Requisito	RF-014
Descripción	La ruta de <i>pathfinding</i> asociada a la tecla '1' llevará al usuario hacia la cafetería.
Prueba	Se comprueba que activando esta ruta, el jugador se desplaza hacia el ítem de cambio de nivel que permite acceder a la escena de la cafetería.
Validación	Correcta

Tabla 55: Tabla de validación para RF-014

Requisito	RF-015
Descripción	La ruta de <i>pathfinding</i> asociada a la tecla '2' llevará al usuario hacia el Punto de Información del Campus.
Prueba	Se comprueba que activando esta ruta, el jugador se desplaza hacia el ítem de cambio de nivel que permite acceder a la escena del PIC.
Validación	Correcta

Tabla 56: Tabla de validación para RF-015

Requisito	RF-016
Descripción	La ruta de <i>pathfinding</i> asociada a la tecla '3' llevará al usuario hacia una de las aulas informáticas genéricas del segundo piso.
Prueba	Se comprueba que activando esta ruta, el jugador se desplaza hacia el ítem de cambio de nivel que permite acceder al primer piso, y de ahí al segundo. Una vez en el segundo piso el jugador se desplaza hacia el ítem de cambio de nivel que permite acceder al aula de informática.
Validación	Aceptable

Tabla 57: Tabla de validación para RF-016

Requisito	RF-017
Descripción	La ruta de <i>pathfinding</i> asociada a la tecla '4' llevará al usuario hacia una de las aulas de clase del tercer piso.
Prueba	Se comprueba que activando esta ruta, el jugador se desplaza hacia el ítem de cambio de nivel que permite acceder al primer piso, luego al segundo y de ahí al tercero. Una vez en el tercer piso el jugador se desplaza hacia el ítem de cambio de nivel que permite acceder al aula de clases.
Validación	Aceptable

Tabla 58: Tabla de validación para RF-017



Requisito	RNF-001
Descripción	La aplicación podrá ejecutarse en varios sistemas operativos además de Windows.
Prueba	Se comprueba si al compilar proyecto para otros sistemas operativos la aplicación funciona correctamente.
Validación	No verificado

Tabla 59: Tabla de validación para RNF-001

Requisito	RNF-002
Descripción	El tiempo de carga entre niveles/pisos deberá ser inferior a dos segundos en un ordenador de gama media actual.
Prueba	Se realiza una estimación de los tiempos de carga.
Validación	Correcta

Tabla 60: Tabla de validación para RNF-002

Requisito	RNF-003
Descripción	La aplicación será capaz de mantener un rendimiento óptimo en ordenadores de variadas características.
Prueba	Se realiza una estimación de velocidad y estabilidad al probar la aplicación en varios ordenadores.
Validación	Correcta

Tabla 61: Tabla de validación para RNF-003

Según se aprecia en las tablas anteriores, todos los requisitos se cumplen correctamente salvo unas escasas excepciones.

El problema que surge durante el *pathfinding* hacia plantas superiores es que el sistema de navegación por mallas que ofrece Unity 3D realiza un mapeo del modelo que se carga en la escena, y en este caso en particular, este mapeo no incluye la zona de los soportales de la planta baja del edificio, lo que no permite realizar un cálculo completo de la ruta hasta los ítems de cambio de nivel correspondientes. El usuario debe acercarse a dicho ítem y activarlo. No se trata, por tanto, de una solución completa de *pathfinding*, pero el sistema sí que realiza los cálculos de ruta para el mapeo disponible, por lo que se puede considerar como una solución aceptable para un prototipo como el que se ha desarrollado.

Como conclusión para este análisis se puede afirmar que la propuesta implementada sigue los requisitos establecidos al comienzo del desarrollo.

## EVALUACIÓN

Para determinar el alcance de la experiencia inmersiva que aporta la aplicación desarrollada, junto a otra serie de factores útiles para futuros cambios o mejoras en la aplicación, se ha creado una pequeña y sencilla encuesta, solicitando a personas anónimas que probasen la aplicación y rellenasen dicha encuesta al final.

Dicha encuesta consta de las siguientes preguntas, a valorar entre 1 y 5, siendo 1 la valoración más baja y 5 la más alta:

- ¿Considera sencilla la dinámica de la aplicación?
- ¿Considera cómodos los controles del jugador?
- Valore la fluidez de la aplicación en y entre los niveles
- Valore el realismo apreciado en los modelos de la aplicación
- Valore la experiencia de inmersión mediante el uso de gafas de realidad virtual

A continuación se muestra un pequeño estudio de los resultados obtenidos. Para cada gráfica, el eje X constituye el número de encuestados y el eje Y constituye la valoración de 1 a 5 para dicha pregunta.

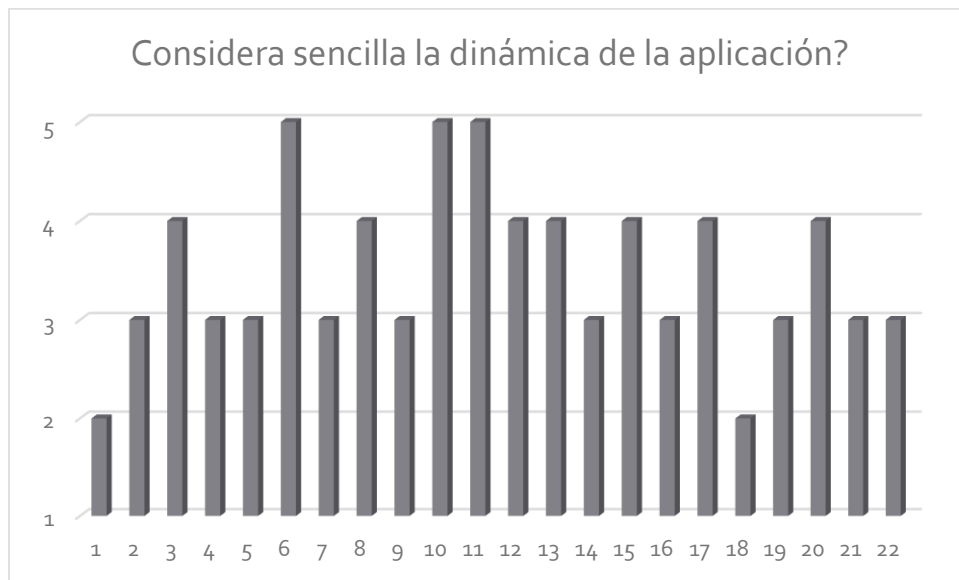


Ilustración 26: Resultados de la primera pregunta de la encuesta

A la primera pregunta, únicamente un 9% de los encuestados opina que el funcionamiento y la dinámica de la aplicación le resultan complicadas.



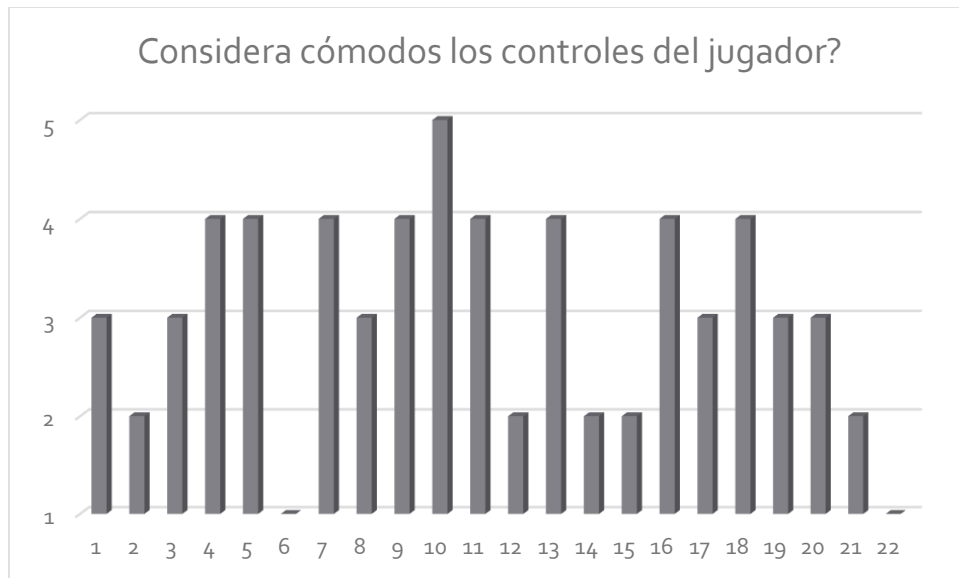


Ilustración 27: Resultados de la segunda pregunta de la encuesta

En esta ocasión existe disparidad de opiniones, donde casi un tercio de los encuestados opina que los controles son poco o nada cómodos. Esto podría deberse a la combinación gafas + teclado, lo que requiere que el usuario permanezca en una posición fija mientras rota la vista con la cabeza.

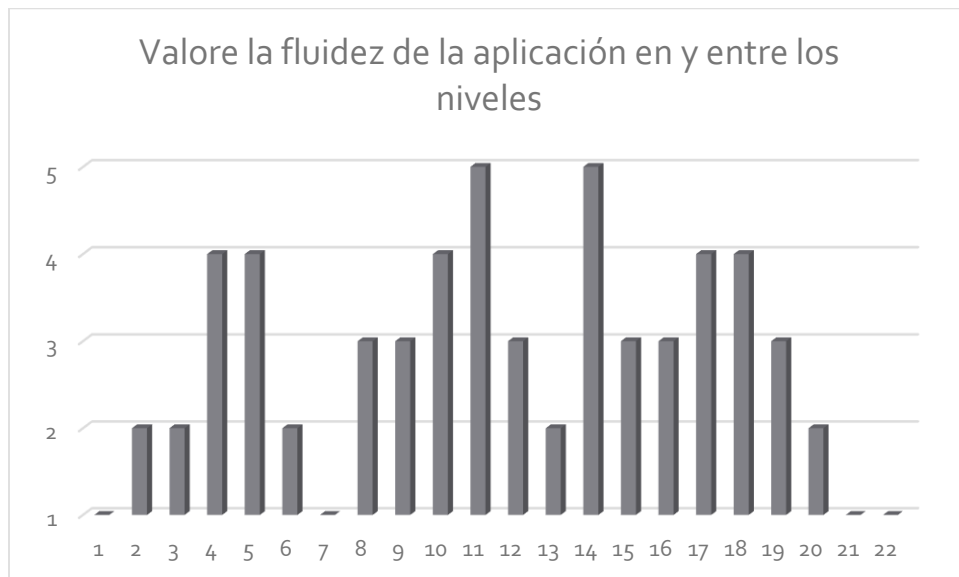


Ilustración 28: Resultados de la tercera pregunta de la encuesta

Un 40% de los encuestados opina que los cambios entre niveles no terminan de ser suficientemente fluidos. Este hecho puede darse por las características del ordenador donde se realizaron las pruebas, porque la aplicación no respondió bien en ese momento o por la alta carga de modelos que tiene este tipo de aplicaciones.

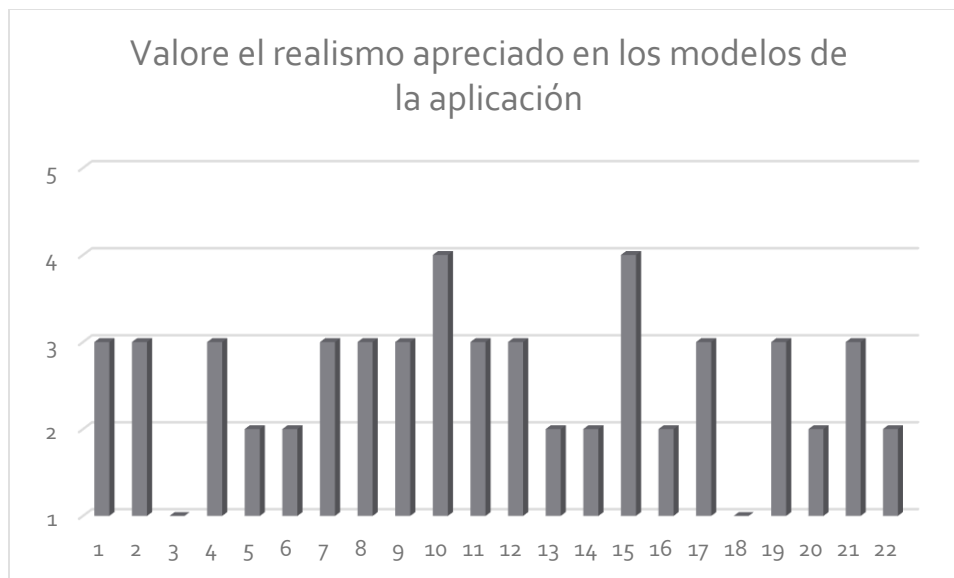


Ilustración 29: Resultados de la cuarta pregunta de la encuesta

De nuevo, un 40% de los encuestados sugiere que los modelos no son demasiado realistas o no están suficientemente detallados. Al tratarse de un prototipo, no se ha querido introducir una gran cantidad de detalles, para que la aplicación pueda ejecutarse con la fluidez más alta posible en todo tipo de ordenadores.

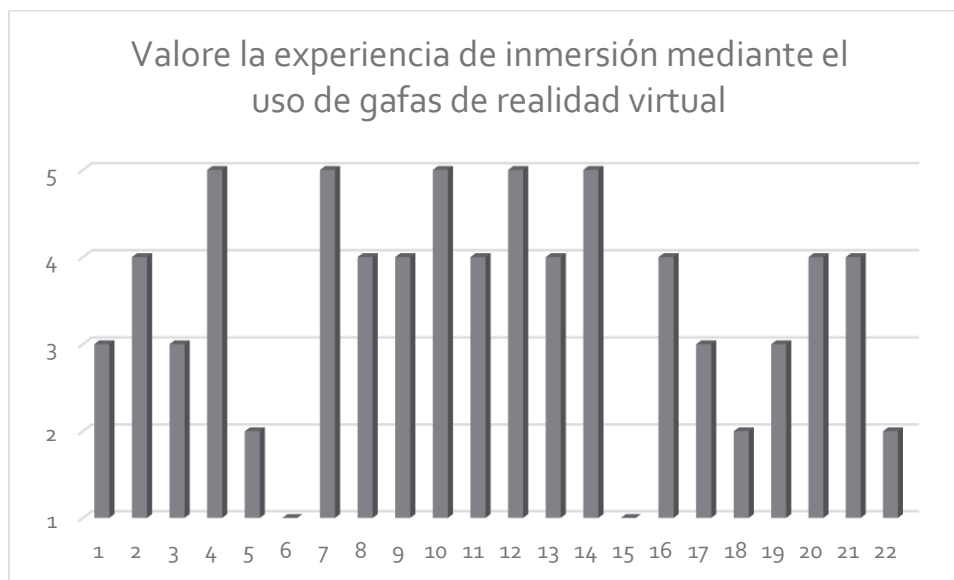


Ilustración 30: Resultados de la quinta pregunta de la encuesta

Por último, en la quinta pregunta, los encuestados opinan, en su gran mayoría, casi un 80%, que la experiencia de realidad virtual y la inmersión que se pretenden están conseguidas. Las bajas valoraciones podrían deberse a mareos o simplemente una mala impresión del usuario que probó la aplicación.



En general el resultado de la encuesta es positivo, considerando que se trata de un prototipo. Las conclusiones que pueden obtenerse de los resultados es que la experiencia inmersiva que se planteó como objetivo principal de este proyecto se ha conseguido, que son necesarios unos controles móviles, como un *gamepad*, en vez de un teclado; y que el realismo es la principal característica a mejorar en el futuro.

## GESTIÓN DEL PROYECTO

En este apartado se exponen las planificaciones estimada y final para el proyecto, junto al presupuesto estimado para el mismo.

### PLANIFICACIÓN INICIAL

Al comienzo de este proyecto, el 22 de abril, se estimó una planificación del tiempo que se tardaría en realizar las múltiples y diferentes tareas de las que consta. Debido a los periodos de clase, exámenes y vacaciones, se fijó una jornada de trabajo estimada de 1 hora de lunes a viernes durante los meses de marzo, abril, mayo y junio (hasta la finalización del periodo de exámenes), de 3 horas durante julio y de 4 horas durante agosto y septiembre. Se calculó que el trabajo estaría terminado para la primera semana de septiembre.

El trabajo se ha dividido en varias fases perfectamente diferenciadas. En primer lugar se realiza la fase de estudio del problema, junto a un análisis y aprendizaje de las tecnologías a utilizar durante el desarrollo. A continuación se realiza la fase de implementación, durante y al final de la cual se harían unas pruebas de la aplicación. Por último se procede a desarrollar la documentación del trabajo realizado.

La fase de estudio, análisis y aprendizaje tiene varios objetivos principales: analizar el problema junto a trabajos realizados y alternativas del mercado para dar con la mejor propuesta o solución, investigar sobre las tecnologías apropiadas para la propuesta y, una vez establecidas las tecnologías, realizar un aprendizaje de las mismas para conocer el entorno de modelado y programación con el que se trabajará.

Durante la fase de implementación se desarrolla la aplicación en su conjunto, pasando primero por la fase de modelado de los niveles y objetos, programando después los niveles, el control del jugador y el movimiento por la aplicación, incluyendo después los módulos de vídeo y de *pathfinding* y, finalmente antes de las pruebas, la compatibilidad con el hardware de realidad virtual.

La primera fase de pruebas se realiza prácticamente de modo simultáneo a la de implementación, pues es necesario comprobar que los scripts y los ajustes que se van desarrollando funcionan apropiadamente. Esta primera fase de pruebas se realiza sin el uso del hardware de realidad virtual.

La segunda fase de pruebas se realiza al final de la implementación, ya con el hardware de VR. En base a los resultados se realizan pequeños ajustes y retoques a los modelos o a objetos y scripts de la fase de programación.

Por último se escribirá la documentación del proyecto. El resultado será un documento entre 40 y 60 páginas, en el que se describirá el trabajo realizado a lo largo del desarrollo del proyecto, evaluando los resultados obtenidos.



El diagrama de Gantt muestra la planificación inicial estimada para el presente proyecto, dividido en las fases comentadas anteriormente (se considera que los periodos que se solapan cuentan como doble sesión de trabajo).

Debido al gran tamaño del gráfico, se incluye como documento adjunto en el apartado “*Anexo III: Diagramas de planificación*”.

## PLANIFICACIÓN FINAL

El diagrama de Gantt muestra la planificación final de tiempo invertido durante el desarrollo de este trabajo de fin de grado (se considera que los periodos que se solapan cuentan como doble sesión de trabajo).

Debido al gran tamaño del gráfico, se incluye como documento adjunto en el apartado “*Anexo III: Diagramas de planificación*”.

El desarrollo del proyecto ha transcurrido por lo general según lo planificado, salvo por una serie de diferencias. En primer lugar, ha de tenerse en cuenta que el alumno se encontraba aún en periodo de clases y de exámenes hasta bien entrado el mes de junio, lo que influyó notoriamente en la demora de la fase de análisis, estudio y aprendizaje, junto a la fase de modelado.

Asimismo, debido a incongruencias en el modelo y algunos problemas a la hora de modelar el edificio y sus niveles, esta fase de modelado duró más de lo previsto inicialmente, solapándose con la fase de implementación, ya que al introducir el modelo en Unity hubo que retocar y arreglar algunas partes del edificio, ya que Blender y el propio Unity perciben de forma diferente el mismo modelo, por motivos de importación o peculiaridades del formato o del propio modelo.

El parón que aparece en la semana del 13 al 20 de agosto se debe a una salida vacacional del alumno, comenzando a su vuelta la implementación de los vídeos y del módulo de *pathfinding*.

Por último, el cierre de la universidad durante todo el mes de agosto condicionó la fase de pruebas con el hardware de realidad virtual, ya que se trata de material del laboratorio y debe permanecer en la universidad.

Pese a estos inconvenientes se considera que la planificación inicial fue correcta, ya que la planificación real se aproxima a la estimada.

## PRESUPUESTO

Este apartado tiene como objetivo el cálculo de un presupuesto estimado para el desarrollo del presente proyecto.

La versión final de la aplicación puede llegar a tener algún cierto valor económico, por lo que resulta necesario crear un presupuesto que permita conocer los gastos asociados al personal y al material amortizable del que se dispone durante el desarrollo.

El coste de los recursos humanos para el proyecto incluye al alumno y a los tutores del trabajo, apareciendo estos últimos como un único ente para el propósito de este presupuesto, y considerando que el número de horas trabajadas por estos es un cuarto de la correspondiente a la del alumno (ofrecen ayuda y asistencia durante el proyecto pero tienen sus propias labores que desarrollar en sus puestos de trabajo).

Se tomará el gasto de personal de la siguiente forma:

- El coste estimado para el alumno, considerado como desarrollador con baja experiencia a efectos prácticos, será de 12 €/hora (24.000 € brutos/año) <sup>[40]</sup>.
- El coste estimado para los tutores, considerados como jefe de proyecto con alta experiencia a efectos prácticos, será de 24 €/hora (48.000 € brutos/año) <sup>[40]</sup>.

El gasto de personal queda, pues, de la siguiente forma:

Gastos alumno		
Fase del desarrollo	Horas trabajadas (h)	Coste (€)
Estudio, análisis y aprendizaje	35	420,00
Implementación y pruebas	253	3036,00
Documentación	44	528,00
<b>Total</b>	<b>332</b>	<b>3984,00</b>
Gastos tutores		
Fase del desarrollo	Horas trabajadas (h)	Coste (€)
Estudio, análisis y aprendizaje	8.75	210,00
Implementación y pruebas	63.25	1518,00
Documentación	11	264
<b>Total</b>	<b>83</b>	<b>1992</b>
<b>Total del personal</b>	<b>415</b>	<b>5976</b>

Tabla 62: Gastos de personal

En la siguiente tabla se muestran los gastos relativos al material amortizable, en los que se encuentran los productos hardware y software utilizados en el proyecto.

Tipo	Artículo	Precio (€)	Amortización (meses)	Amortización (€/mes)	Periodo de uso (meses)	Coste en proyecto (€)
Hardware	Ordenador de sobremesa para desarrollo	880.00	60	14.70	5	73.50
	Ordenador portátil para pruebas	750.00	60	12.50	5	62.5
	Oculus Rift DK2	275.00	24	11.50	1	11.50
Software	Unity 3D Pro	60.00/mes	-	-	2	120.00
	Blender	0	-	-	-	0
	Microsoft Office 2013	539.00	48	11.25	5	56.25
<b>Total</b>						<b>323.75</b>

Tabla 63: Gastos de material amortizable

Los gastos directos, compuestos por los totales de las tablas anteriores, son 6299,75 €, mientras que los gastos indirectos, como gastos de luz y agua, se estiman en torno al 5% de los gastos directos. Por otra parte, en caso de que el trabajo se vendiese, el margen de beneficios previsto sería del 5% de los gastos. El presupuesto total queda de la siguiente forma:

Concepto	Coste (€)
Gastos directos	6299,75
Gastos indirectos	315
<b>Total del presupuesto</b>	<b>6614,75</b>

Tabla 64: Presupuesto total del proyecto

## CONCLUSIONES

Este apartado constituye las conclusiones finales sobre el desarrollo del presente trabajo de fin de grado. Asimismo se incluyen unas líneas finales sobre las diferentes mejoras que pueden aplicarse sobre la aplicación para favorecer una mayor experiencia de usuario.

## COMENTARIOS FINALES

El objetivo que se planteó para el proyecto al comienzo de su desarrollo fue diseñar y crear una aplicación de visita virtual al edificio Sabatini de la universidad, la cual, mediante el uso de hardware de realidad virtual, ofreciese al usuario una experiencia más inmersiva que la del resto de aplicaciones similares del mercado. Tras concluir el desarrollo de la aplicación y realizar una serie de pruebas, entre las que se ha incluido una encuesta anónima, se puede decir que, pese a no ofrecer el realismo que aportan las fotografías de alta calidad o los vídeos de alta resolución de las posibles alternativas, la experiencia del usuario junto a la inmersión del mismo en el entorno es superior a la que ofrece la competencia, lo que permite afirmar que se ha conseguido cumplir el objetivo principal que se estableció en un principio.

Respecto al proceso completo del desarrollo, se trata de un proyecto complejo, con muchas tareas que realizar, siendo además todas ellas de carácter muy diverso, lo que sin duda ha puesto a prueba al alumno para realizar una labor de investigación y aprendizaje de gran calibre, que ha supuesto muchas semanas de trabajo, con el añadido de haber comprendido los meses de verano, donde se acostumbra a desconectar del curso académico anterior.

El desarrollo de este proyecto ha sido, sin duda alguna, de gran ayuda para reforzar capacidades personales de trabajo, como la redacción formal de textos, la planificación de tareas o la organización de las mismas, así como capacidades técnicas relacionadas con la gran cantidad de conocimientos adquiridos durante las asignaturas de la carrera. Tras observar los resultados obtenidos, el alumno posee los conocimientos necesarios para iniciar o participar en proyectos de mayor envergadura que el realizado durante estos meses.



## LÍNEAS FUTURAS

Existen muchas mejoras, tanto técnicas como gráficas, que pueden ser desarrolladas para ofrecer una experiencia al usuario aún mejor. Por ejemplo, los modelos de la aplicación no son todo lo realistas que se podría desear, lo que puede llevar a que muchos posibles usuarios o clientes sean reticentes a adquirir la aplicación, puesto que ofrece un aspecto más 'arcade' que un simulador corriente, tomando algunos matices de videojuego que a los usuarios más puristas o exigentes puede no gustarles. En este sentido, realizar una remodelación del edificio o incluir un modelo más 'limpio' serían las opciones más adecuadas para alcanzar un mayor realismo.

Por otra parte, el módulo de *pathfinding* puede ser mejorado, añadiendo nuevos puntos de destino, pero sobre todo mejorando el mapeo realizado por el sistema de navegación por mallas que ofrece Unity. Además, puede habilitarse un tipo de control diferente utilizando, por ejemplo, un *gamepad* compatible con el sistema operativo.

A su vez, la aplicación podría mejorar la experiencia completa del usuario alterando el sistema de niveles actual por un modelado completo del edificio que conste de un único nivel base por el que el usuario pueda desplazarse sin necesitar cargar los pisos superiores como niveles adicionales, y que incluya, por ejemplo, las zonas de escaleras y baños, que no aparecen modeladas en esta versión de la aplicación por motivos de rendimiento. Esto podría conseguirse utilizando un equipo de desarrollo más potente junto a un motor gráfico alternativo como Unreal Engine 4.

Por último, pueden añadirse funcionalidades adicionales a los niveles POI, como una galería interactiva de fotos, navegación entre vídeos (que posibilitaría la inclusión de más archivos de vídeo entre los que elegir), etc.

## REFERENCIAS

- [1] Michael W. McGreevy, "The Virtual Environment Display System", NASA Archive, 1985, <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19910013709.pdf>
- [2] Valentina Palladino, "Specs Sheet: Sony's Project Morpheus takes Oculus Rift head-on", The Verge, 2014, <http://www.theverge.com/2014/3/21/5533914/spec-sheet-sonys-project-morpheus-takes-on-oculus-rift>
- [3] Avegant, "Glyph: A Mobile Personal Theater With Built In Premium Audio", Kickstarter, <https://www.kickstarter.com/projects/avegantglyph/a-mobile-personal-theater-with-built-in-premium-au>
- [4] Dean Takahashi, "Avegant Glyph virtual retinal display is like shining an 80-inch TV image on your eyeballs", VentureBeat, Enero 2014, <http://venturebeat.com/2014/01/06/avegant-glyph-retina-display-is-like-shining-an-80-inch-tv-image-on-your-eyeballs/>
- [5] Paul James, "The Oculus Rift DK2, In-Depth Review and DK1 Comparison", Road to VR, Julio 2014, <http://www.roadtovr.com/oculus-rift-dk2-review-dk1-comparison-vr-headset/>
- [6] Jamie O'Flanagan, "Game Engine Analysis and Comparison", Gamesparks, 2014, <http://www.gamesparks.com/blog/game-engine-analysis-and-comparison/>
- [7] Santiago Tejedor, "¿Realidad virtual y videojuegos en 2014?", Micromanía, Marzo 2014, <http://www.micromania.es/tech/realidad-virtual-2014>
- [8] Anón. , "Historia de la realidad virtual", Dpto. de Información y Comunicaciones, Universidad de la Coruña, <http://sabia.tic.udc.es/gc/Contenidos%2oadicionales/trabajos/3D/Realidad%2oVirtual/web/historia.html>
- [9] Scott Fisher quote, "NASA-Ames", From Wagner to Virtual Reality, [http://www.w2vr.com/archives/Fisher/04\\_NASA.html](http://www.w2vr.com/archives/Fisher/04_NASA.html)
- [10] Paul James, "InfinitEye 210 Degree HMD Technical Q&A: How does High FOV Virtual Reality Work?", Road to VR, Noviembre 2013, <http://www.roadtovr.com/infiteye-technical-ga-high-fov-virtual-reality-work/>
- [11] Keith Stuart, "Sony: Project Morpheus virtual reality headset is a significant investment", The Guardian, Julio 2014, <http://www.theguardian.com/technology/2014/jul/02/sony-project-morpheus-virtual-reality-headset>
- [12] Kirk Hamilton, "We tried Sony's PS4 Virtual Reality Headset. We Like It.", Kotaku, Marzo 2014, <http://kotaku.com/we-just-tried-sonys-ps4-virtual-reality-headset-we-l-1547417858>
- [13] Unity Manual, <http://docs.unity3d.com/Manual/index.html>
- [14] Unity 3D Scripting API, <http://docs.unity3d.com/ScriptReference/index.html>



- [15] *Blender Manual* v2.6, <http://wiki.blender.org/index.php/Doc:ES/2.6/Manual>
- [16] Mark Masters, "Unity, UDK, Unreal Engine 4 or CryENGINE – Which Game Engine Should I Choose?", Digital-Tutors, Enero 2014, <http://blog.digitaltutors.com/unity-udk-cryengine-game-engine-choose/>
- [17] *Estereoscopio*, Historia de la realidad virtual, <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/Realidad%20Virtual/web/historia.html>
- [18] *Gafas 3D*, Xataka, <http://img.xataka.com/2011/03/dolby-3d-gafas-2011.jpg>
- [19] *Sistema de anaglifos*, NextGenRDI, [http://4.bp.blogspot.com/-aS4EN6Jc154/T4RjO-zOmxI/AAAAAAAAAkQ/m3o3YNLFKDM/s1600/Colour3D\\_2.PNG](http://4.bp.blogspot.com/-aS4EN6Jc154/T4RjO-zOmxI/AAAAAAAAAkQ/m3o3YNLFKDM/s1600/Colour3D_2.PNG)
- [20] *Cartel promocional de Sensorama*, Proyecto IDIS, <http://proyectoidis.org/wp-content/uploads/2013/07/sensorama01.jpg>
- [21] *Grimes glove*, Ohio State University, <https://design.osu.edu/carlson/history/images/small/grimes-glove.jpg>
- [22] *VIVED*, High-Tech History, <http://hightechhistory.files.wordpress.com/2012/01/vivedo8.gif>
- [23] *Casco VR NASA-Ames*, Zakros, <http://www.zakros.com/ucb/histS99/Notes/Class6/Fisher.HMD.jpeg>
- [24] *Nintendo Virtual Boy*, The Plan Company, [http://www.theplancompany.com/wp-content/uploads/2014/03/virtual\\_boy.jpg](http://www.theplancompany.com/wp-content/uploads/2014/03/virtual_boy.jpg)
- [25] *SEGA Virtual Reality*, Born to Play, <http://www.borntoplay.es/wp-content/uploads/2014/03/segavr.jpg>
- [26] *Nintendo Wii*, DeCulture, <http://www.deculture.es/wp-content/uploads/2011/04/Wii.jpg>
- [27] *Microsoft Kinect*, Meristation, [http://img3.meristation.com/files/imagenes/general/kinect\\_2.jpg](http://img3.meristation.com/files/imagenes/general/kinect_2.jpg)
- [28] *Oculus Rift DK1*, OculusVR, <https://dbvc4uanumi2d.cloudfront.net/cdn/4.0.9/wp-content/themes/oculus/img/order/dk1-product.jpg>
- [29] *Bioshock Infinite 'City in the sky' Screenshot*, Bioshock Infinite Official Page, [http://www.bioshockinfinite.com/images/media/screenshots/cityinthesky\\_ONLINE\\_wideuse.jpg](http://www.bioshockinfinite.com/images/media/screenshots/cityinthesky_ONLINE_wideuse.jpg)
- [30] *Project Morpheus*, PlayStation EU Blog, [http://farm8.staticflickr.com/7259/13248110323\\_58807b59ac\\_b.jpg](http://farm8.staticflickr.com/7259/13248110323_58807b59ac_b.jpg)
- [31] *Avegant Glyph*, Avegant Official Page, [http://avegant.com/wp-content/themes/avegant/images/color\\_grey.png](http://avegant.com/wp-content/themes/avegant/images/color_grey.png)
- [32] *Avegant Glyph Virtual Retinal Display*, Life Really Matters, <http://lifereallymatters.com/wp-content/uploads/2014/03/Glyph-Tech.png>



- [33] *InfiniteEye*, Road to VR, <http://i2.wp.com/www.roadtovr.com/wp-content/uploads/2013/11/infiniteeye-inside-large.jpg>
- [34] *Oculus Rift DK1*, John Rockefeller, <http://www.johnrockefeller.net/wp-content/uploads/2014/03/oculus-rift-dk1.jpg>
- [35] *Oculus Rift DK2*, OculusVR, <https://dbvc4uanumi2d.cloudfront.net/cdn/4.0.9/wp-content/themes/oculus/img/order/dk2-product.jpg>
- [36] *Screen Door Effect Oculus Rift Comparison*, OculusVR Forums, [http://horobox.co.uk/u/orkel\\_1406685575.jpg](http://horobox.co.uk/u/orkel_1406685575.jpg)
- [37] *Unity3D User Interface*, Unity Japan Official Blog, <http://japan.unity3d.com/blog/wp-content/uploads/GUI21.png>
- [38] *Unreal Engine 4 User Interface*, Gamesparks, <http://www.gamesparks.com/wp-content/uploads/2014/06/Unreal-layout-1024x574.jpg>
- [39] *CryENGINE User Interface*, CryENGINE Documentation, <http://docs.cryengine.com/download/attachments/1310910/TriggerFG.jpg?version=1&modificationDate=1360842711000&api=v2>
- [40] *Tendencia de salario para ingenieros informáticos*, Infojobs Trends, <http://plandecarrera.infojobs.net/puesto-de-trabajo/ingeniero-informatico>



## ANEXO I: GUÍA RÁPIDA DE CONFIGURACIÓN DEL SISTEMA

En este documento de referencia se pretende especificar los pasos necesarios que debe seguir el usuario de la aplicación para configurar el entorno necesario para la ejecución del paseo virtual, así como los diferentes controles para la aplicación.

Para poder utilizar el sistema se debe disponer, en primer lugar, de la **aplicación completa**, según se distribuye al público. A su vez, es **imprescindible** contar con el hardware de realidad virtual **Oculus Rift**, preferiblemente la versión **DK2**, ya que la compatibilidad de esta versión con la aplicación es superior a la del DK1.

Una vez se disponga de lo anteriormente mencionado, será necesario instalar las Oculus Rift en el sistema operativo. Para la instalación completa del hardware de realidad virtual, consulte la Guía Rápida para Oculus Rift DK2 en el siguiente enlace:

[http://static.oculusvr.com/sdk-downloads/documents/Oculus\\_Rift\\_DK2\\_Instruction\\_Manual.pdf](http://static.oculusvr.com/sdk-downloads/documents/Oculus_Rift_DK2_Instruction_Manual.pdf)

Con las Oculus Rift instaladas y funcionando correctamente, abra la aplicación mediante doble click en el ejecutable de la misma, y cuando cargue el menú principal, seleccione la opción 'Comenzar'.

Cuando cargue la parte exterior del edificio Sabatini, dispondrá de los siguientes controles de teclado para navegar por la aplicación:

- Desplácese por el entorno mediante las teclas :
  - 'W' o la flecha de dirección arriba le permiten andar hacia delante.
  - 'A' o la flecha de dirección izquierda le permiten andar lateralmente hacia la izquierda.
  - 'S' o la flecha de dirección abajo le permiten andar hacia atrás.
  - 'D' o la flecha de dirección derecha le permiten andar lateralmente hacia la derecha.
- Muévase entre los diferentes pisos del edificio mediante las teclas:
  - 'F' le permitirá acceder a los niveles superiores o inferiores del edificio, según proceda.
  - 'G' le permitirá acceder a los puntos de interés del edificio.

Permanezca atento a los objetos flotantes que aparecen en el nivel, le permitirán subir o bajar de planta y acceder a las aulas o a los puntos de interés del edificio.

Gracias al hardware de realidad virtual podrá observar el entorno y desplazarse según la posición hacia la que esté mirando, por lo que no necesitará del uso de ratón para cambiar la cámara.

Mediante el módulo de navegación automática de la aplicación, podrá desplazarse automáticamente siguiendo la ruta calculada por el sistema. Esto le ayudará a alcanzar los siguientes puntos del edificio:

- La tecla numérica '1' le llevará a visitar la cafetería del edificio Sabatini.
- La tecla numérica '2' le llevará a visitar el Punto de Información del Campus (PIC).
- La tecla numérica '3' le llevará a visitar las aulas informáticas del segundo piso.
- La tecla numérica '4' le llevará a visitar las aulas de clase del tercer piso.

Cuando desee salir de la aplicación, basta con que pulse la tecla 'Escape' de su teclado.



## ANEXO II: DIAGRAMAS DE CLASES

A continuación se muestran los diagramas de clases para la solución propuesta:

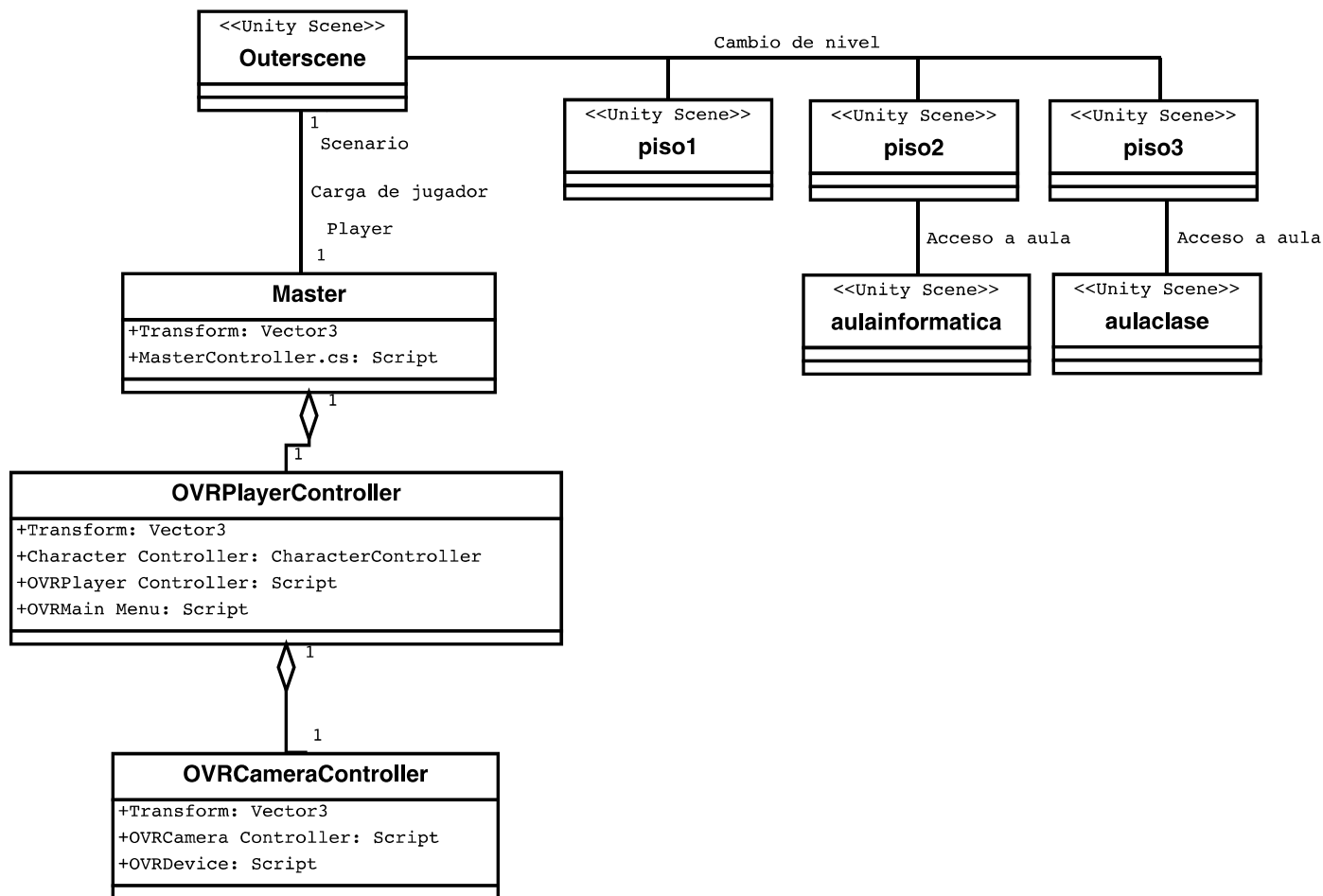


Ilustración 31: Diagrama de clases de relación jugador-niveles



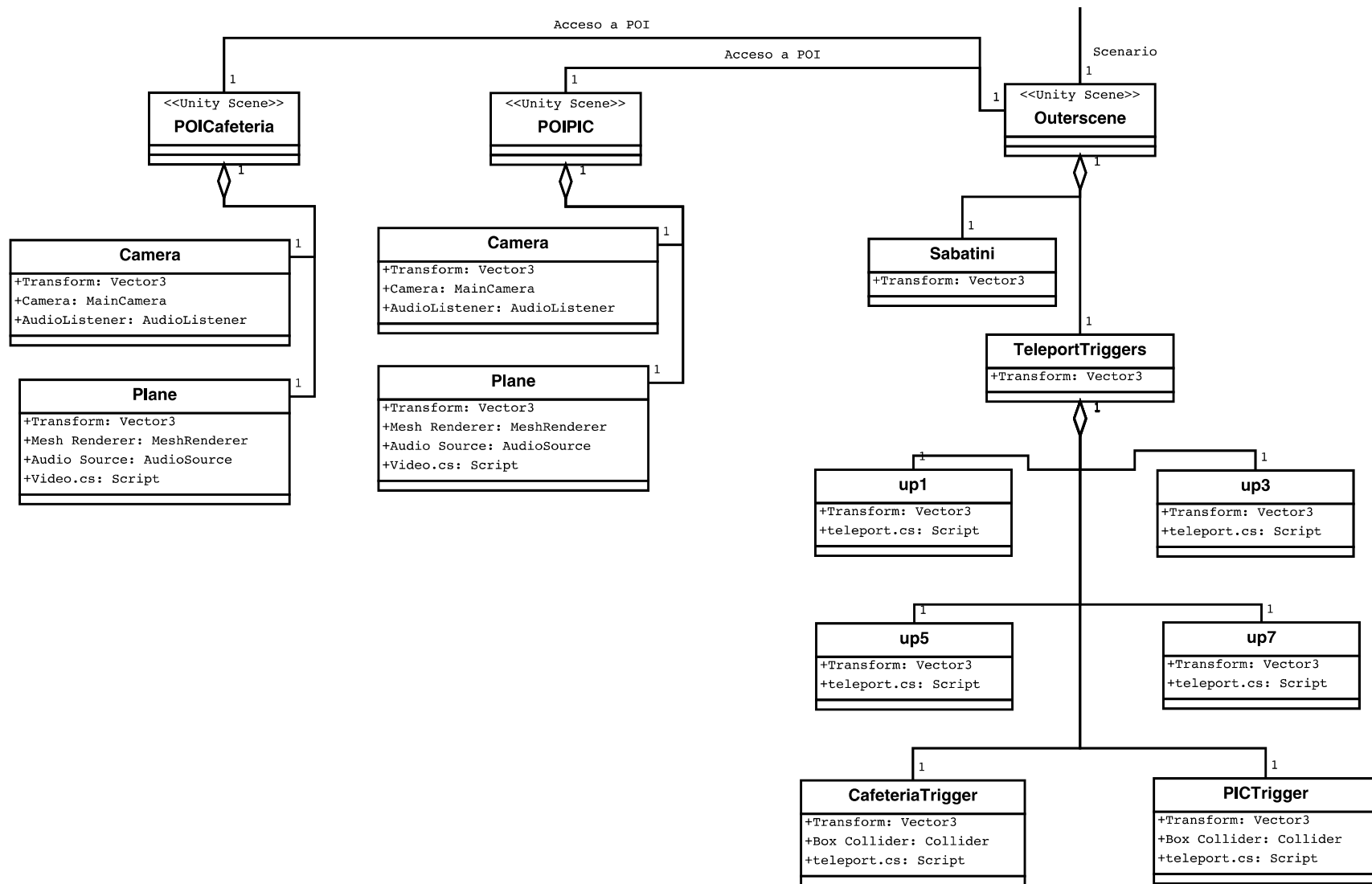


Ilustración 32: Diagrama de clases de relación Outerscene-POI

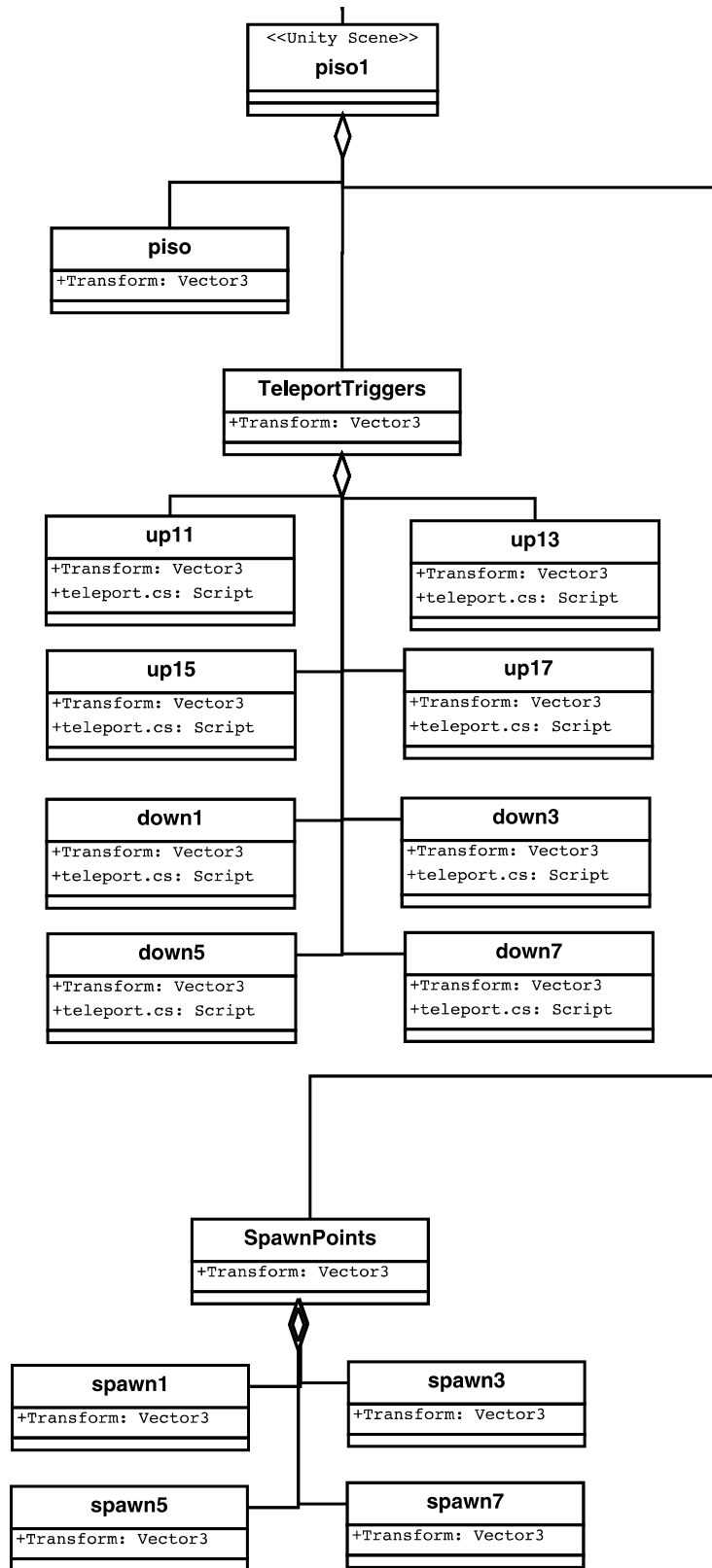


Ilustración 33: Diagrama de clases para piso1

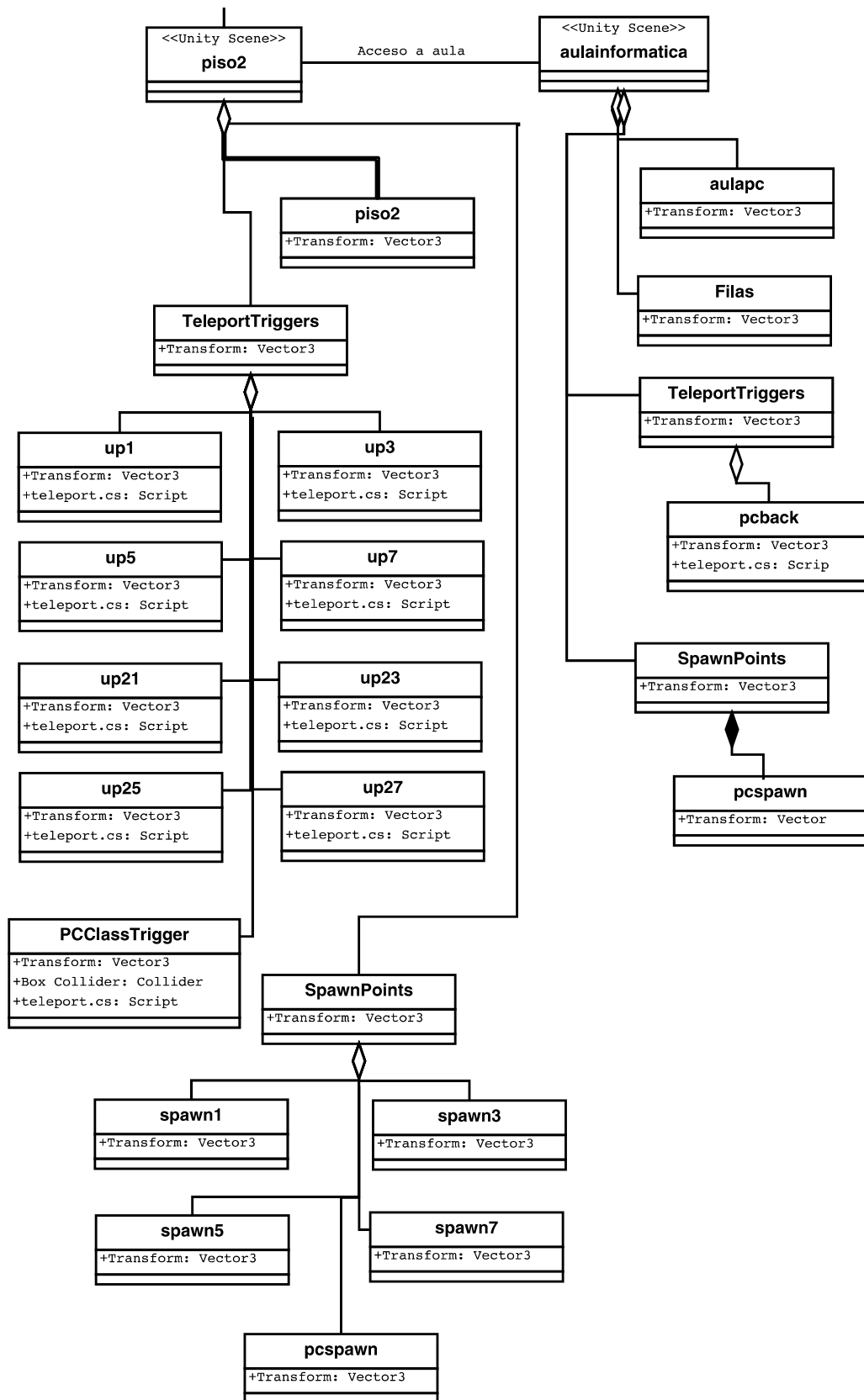


Ilustración 34: Diagrama de clases de la relación piso2-aulainformatica

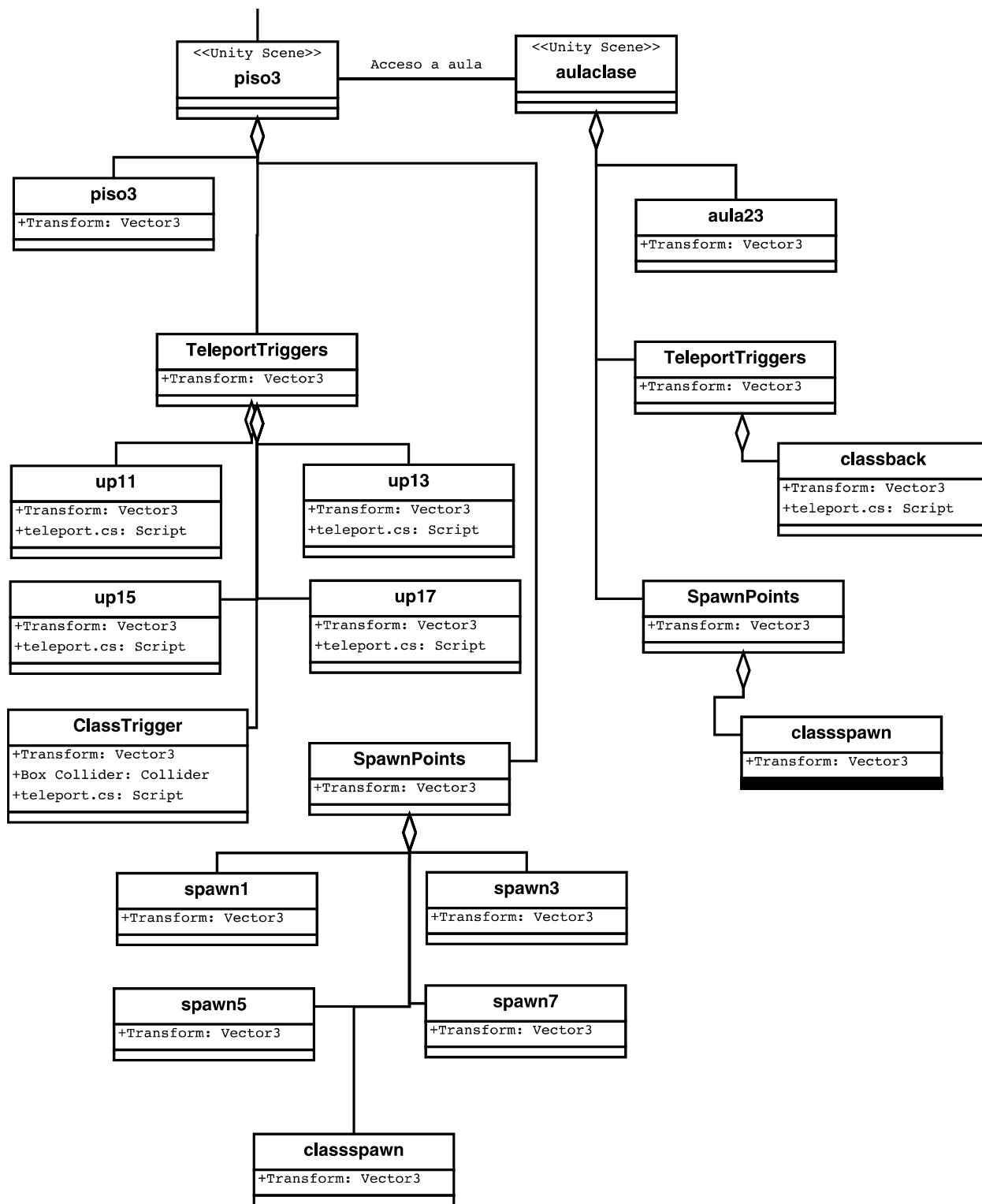


Ilustración 35: Diagrama de clases de la relación piso3-aulaclase



## ANEXO III: DIAGRAMAS DE PLANIFICACIÓN

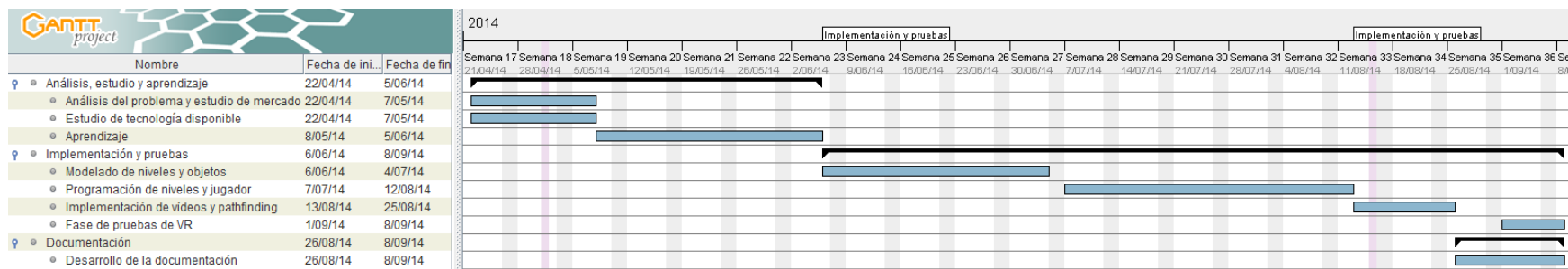


Ilustración 36: Diagrama de planificación inicial estimada

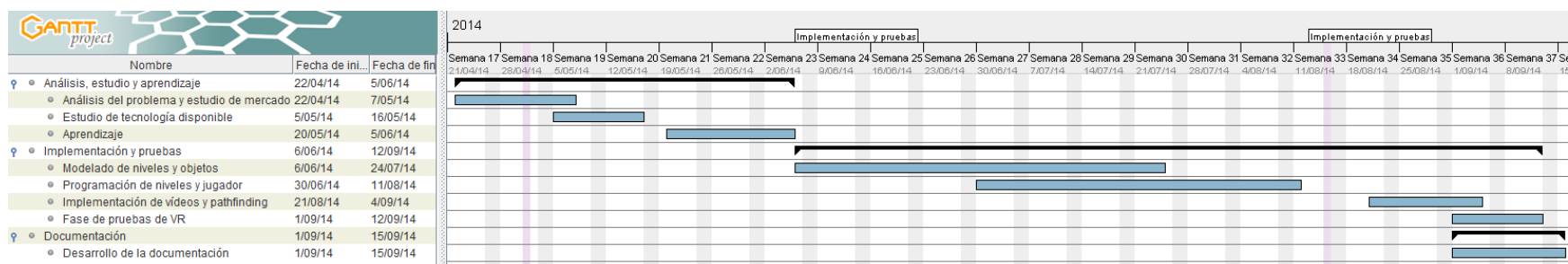


Ilustración 37: Diagrama de planificación final